**TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÍ HỌC SINH**

Sinh viên thực hiện: **TRẦN THỊ KIM OANH – 19IT3**

Giảng viên hướng dẫn: **Ths. NGUYỄN ANH TUẤN**

***Đà Nẵng, tháng 05 năm 2021***

**TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÍ HỌC SINH**

Sinh viên thực hiện: **TRẦN THỊ KIM OANH – 19IT3**

Giảng viên hướng dẫn: **Ths. NGUYỄN ANH TUẤN**

***Đà Nẵng, tháng 05 năm 2021***

# MỤC LỤC

[MỤC LỤC 3](#_2et92p0)

[DANH MỤC HÌNH 6](#_3dy6vkm)

[DANH MỤC BẢNG 8](#_1t3h5sf)

[DANH MỤC VIẾT TẮT 9](#_4d34og8)

[MỞ ĐẦU 10](#_2s8eyo1)

[LỜI CẢM ƠN 11](#_17dp8vu)

[CHƯƠNG 1. TỔNG QUAN 12](#_26in1rg)

[1.1. Lý do chọn đề tài 12](#_lnxbz9)

[1.2. Mục tiêu nghiên cứu 13](#_35nkun2)

[1.3. Nhiệm vụ nghiên cứu 13](#_1ksv4uv)

[1.4. Phương pháp và phạm vi nghiên cứu 13](#_44sinio)

[1.5. Phương tiện nghiên cứu 14](#_2jxsxqh)

[1.6. Xác định những yêu cầu hệ thống 14](#_z337ya)

[1.6.1. Yêu cầu chức năng 14](#_3j2qqm3)

[1.6.2. Yêu cầu phi chức năng 14](#_1y810tw)

[1.7. Kết quả đề tài 15](#_2xcytpi)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 16](#_1ci93xb)

[2.1. Công cụ thiết kế ứng dụng 16](#_3whwml4)

[2.1.1. Android studio 16](#_2bn6wsx)

[2.1.2. Sublime Text 3 17](#_3as4poj)

[2.1.3. Phần mềm phân tích thiết kế hệ thống StarUML 17](#_49x2ik5)

[2.1.4. Công cụ quản trị cơ sở dữ liệu PhpMyAdmin 18](#_147n2zr)

[2.2. Ngôn ngữ thiết kế ứng dụng 18](#_3o7alnk)

[2.2.1. Ngôn ngữ chính lập trình ứng dụng android 18](#_23ckvvd)

[2.2.2. Ngôn ngữ XML (eXtensible Markup Language) 20](#_32hioqz)

[2.2.3. Ngôn ngữ thiết kế hệ thống UML 21](#_2grqrue)

[2.2.4. Ngôn ngữ lập trình PHP 22](#_3fwokq0)

[2.2.2. Framework Laravel 23](#_4f1mdlm)

[2.3. Công nghệ xây dựng ứng dụng 24](#_19c6y18)

[2.3.1. Khái niệm về API RESTful API trong Moblie App 24](#_3tbugp1)

[2.3.2. Khái niệm và cấu trúc json 26](#_37m2jsg)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 32](#_1mrcu09)

[3.1. Biểu đồ Usecase 32](#_46r0co2)

[3.1.1. Biểu đồ Usecase tổng quát 32](#_2lwamvv)

[3.1.2. Biểu đồ Usecase phân rã actor “Giáo Viên” 33](#_3l18frh)

[3.1.3. Biểu đồ Usecase phân rã actor “Học Sinh” 33](#_4k668n3)

[3.1.4. Biểu đồ Usecase phân rã actor “Admin” 34](#_1egqt2p)

[3.1.5. Usecase quản lí danh sách học sinh 34](#_2dlolyb)

[3.1.6. Usecase quản lí điểm 35](#_3cqmetx)

[3.1.7. Usecase quản lí môn học 35](#_4bvk7pj)

[3.1.8. Usecase quản lí lịch học 36](#_1664s55)

[3.1.9. Usecase quản lí và hiệu chỉnh tài khoản. 36](#_25b2l0r)

[3.1.10. Usecase quản lí giáo viên. 37](#_34g0dwd)

[3.2. Đặc tả Usecase 37](#_43ky6rz)

[3.2.1. Đặc Usecase đăng nhập hệ thống 37](#_2iq8gzs)

[3.2.2. Đặc tả Usecase quản lí và hiệu chỉnh thông tin học sinh 38](#_xvir7l)

[3.2.3. Đặc tả Usecase quản lý môn học 39](#_3hv69ve)

[3.2.4. Đặc tả Usecase quản lý Điểm 41](#_1x0gk37)

[3.2.5. Đặc tả Usecase thống kê 43](#_4h042r0)

[3.2.6. Đặc tả Usecase tra cứu thông tin 44](#_2w5ecyt)

[3.2.7. Đặc tả Usecase quản lí giáo viên: 44](#_1baon6m)

[3.2.8. Đặc tả Usecase quản lý và hiệu chỉnh tài khoản. 46](#_3vac5uf)

[3.3. Biểu đồ lớp 49](#_2afmg28)

[CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG 50](#_39kk8xu)

[4.1. Hệ thống CSDL 50](#_1opuj5n)

[4.1.1. Bảng user 50](#_48pi1tg)

[4.1.2. Bảng giáo viên 50](#_1302m92)

[4.1.3. Bảng học sinh 51](#_2250f4o)

[4.1.4. Bảng lớp 51](#_319y80a)

[4.1.5. Bảng lịch học 51](#_40ew0vw)

[4.1.6. Bảng môn học 52](#_upglbi)

[4.2. Thiết kế ứng dụng 53](#_1tuee74)

[4.2.1. Phần web API 53](#_4du1wux)

[4.2.2. Phần ứng dụng Android 55](#_36ei31r)

[KẾT LUẬN 65](#_rjefff)

[Kết quả đạt được 65](#_3bj1y38)

[Hướng phát triển 66](#_1qoc8b1)

[TÀI LIỆU THAM KHẢO 67](#_4anzqyu)

# 

# DANH MỤC HÌNH

[Hình 2 - 1: Công cụ lập trình Android Studio 15](#_qsh70q)

[Hình 2 - 2: Công cụ lập trình Sublime Text 3 16](#_1pxezwc)

[Hình 2 - 3: Phần mềm phân tích thiết kế hệ thống StarUML 17](#_2p2csry)

[Hình 2 - 4: Ngôn ngữ lập trình Java 18](#_ihv636)

[Hình 2 - 5: Ngôn ngữ XML(eXtensible Markup Language) 19](#_1hmsyys)

[Hình 2 - 6: Ngôn ngữ thiết kế hệ thống UML 20](#_vx1227)

[Hình 2 - 7: Ngôn ngữ lập trình PHP 21](#_1v1yuxt)

[Hình 2 - 8: Cấu trúc mô hình MVC 23](#_2u6wntf)

[Hình 2 - 9: Mô hình RESTful API 25](#_28h4qwu)

[Hình 2 - 10: Mô hình API chi tiết 25](#_nmf14n)

[Hình 3 - 1: Biểu đồ Use-case tổng quát 31](#_111kx3o)

[Hình 3 - 2: Biểu đồ Usecase phân rã actor “Giáo Viên” 32](#_206ipza)

[Hình 3 - 3: Biểu đồ Usecase phân rã actor “Học Sinh” 32](#_2zbgiuw)

[Hình 3 - 4: Biểu đồ Usecase phân rã actor “Admin” 33](#_3ygebqi)

[Hình 3 - 5: Usecase quản lý học sinh 33](#_sqyw64)

[Hình 3 - 6: Usecase quản lí điểm 34](#_1rvwp1q)

[Hình 3 - 7: Usecase quản lí môn học 34](#_2r0uhxc)

[Hình 3 - 8: Usecase quản lí lịch học 35](#_3q5sasy)

[Hình 3 - 9: Usecase quản lí và hiệu chỉnh tài khoản. 35](#_kgcv8k)

[Hình 3 - 10: Usecase quản lí giáo viên. 36](#_1jlao46)

[Hình 3 - 11: Biểu đồ lớp 48](#_pkwqa1)

[Hình 4 - 1: Trang chủ website 53](#_2szc72q)

[Hình 4 - 2: Danh sách học sinh xếp theo lớp 54](#_184mhaj)

[Hình 4 - 3: Danh sách môn học 54](#_3s49zyc)

[Hình 4 - 4: Danh sách lịch học 55](#_279ka65)

[Hình 4 - 5: Danh sách khen thưởng 55](#_meukdy)

[Hình 4 - 6: Giao diện đăng nhập ứng dụng 56](#_1ljsd9k)

[Hình 4 - 7: Giao diện trang chủ ứng dụng 57](#_45jfvxd)

[Hình 4 - 8: Giao diện danh sách học sinh phân theo lớp 58](#_2koq656)

[Hình 4 - 9: Thêm học sinh 59](#_zu0gcz)

[Hình 4 - 10: Chỉnh sửa và cập nhật thông tin học sinh 59](#_3jtnz0s)

[Hình 4 - 11: Xóa học sinh ra khỏi danh sách 60](#_1yyy98l)

[Hình 4 - 12: Danh sách môn học 61](#_4iylrwe)

[Hình 4 - 13: Danh sách khen thưởng 62](#_2y3w247)

[Hình 4 - 14: Giao diện lịch học 63](#_1d96cc0)

[Hình 4 - 15: Giao diện thay đổi mật khẩu người dùng 63](#_3x8tuzt)

[Hình 4 - 16: Giao diện thông tin tài khoản, đăng xuất 64](#_2ce457m)

# DANH MỤC BẢNG

[Bảng 2 - 1: So sánh XML với HTML 20](#_41mghml)

[Bảng 4 - 1: Bảng user 50](#_2nusc19)

[Bảng 4 - 2: Bảng giáo viên 50](#_3mzq4wv)

[Bảng 4 - 3: Bảng học sinh 51](#_haapch)

[Bảng 4 - 4: Bảng lớp 51](#_1gf8i83)

[Bảng 4 - 5: Bảng lịch học 52](#_2fk6b3p)

[Bảng 4 - 6: Bảng môn học 52](#_3ep43zb)

# DANH MỤC VIẾT TẮT

| **STT** | **Viết Tắt** | **Nội Dung** |
| --- | --- | --- |
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | IDI | Integrated Development Environment  ( Môi trường phát triển tích hợp) |
| 3 | UML | Unified Modeling Language  (Ngôn ngữ mô hình hóa thống nhất) |
| 4 | MDA | Model Driven Architecture  (Kiến trúc định hướng mô hình) |
| 5 | RDBMS | Relational Database Management System  **(**Hệ quản trị cơ sỡ dữ liệu quan hệ) |
| 6 | XML | eXtensible Markup Language  (ngôn ngữ đánh dấu mở rộng) |
| 7 | SGML | Standard Generalized Markup Language  ( Ngôn ngữ tổ chức và gắn thẻ yếu tố của một tài liệu) |
| 8 | HTML | Hyper Text Markup Language  ( Ngôn ngữ đánh dấu siêu văn bản) |
| 9 | PHP | Hypertext Preprocessor  (Ngôn ngữ lập trình kịch bản) |
| 10 | UC | UseCase |
| 11 | API | Application Programming Interface  (Giao diện lập trình ứng dụng) |

# MỞ ĐẦU

Cùng với sự phát triển không ngừng của các ngành khoa học kỹ thuật, các ngành công nghiệp cũng phát triển nhanh chóng. Việc áp dụng những công nghệ hiện đại vào nhiều lĩnh vực trong cuộc sống đó là điều tất yếu. Song song với sự phát triển đó là sự phát triển về văn hóa, giáo dục.

Sự ra đời của công nghệ thông tin là sự tích hợp đồng thời các tiến bộ về công nghệ và tổ chức thông tin, đem đến nhiều ảnh hưởng tích cực cho sự phát triển của giáo dục. Công nghệ thông tin và đặc biệt là sự phát triển của internet mở ra một kho kiến thức vô cùng đa dạng và phong phú cho người học và người dạy, giúp cho việc tìm hiểu kiến thức đơn giản hơn rất nhiều, cải thiện chất lượng học và dạy. Chính vì thế và việc áp dụng công nghệ thông tin vào quản lí trường học không phải là một điều gì xa lạ đối với chúng ta nhưng áp dụng như nào là đúng và hợp lý thì không phải trường học nào cũng làm được.

Do đó, nhóm chúng em quyết định thực hiện đề tài: “Xây dựng ứng dụng quản lí học sinh”. Ứng dụng sẽ giúp giáo viên có thể quản lý học sinh một cách dễ dàng và tối ưu nhất.

# LỜI CẢM ƠN

Trong suốt thời gian làm đồ án, nhóm chúng em nhận được rất nhiều sự giúp đỡ. Chúng em xin chân thành cảm ơn Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã tạo điều kiện thuận lợi cho em thực hiện đề tài đồ án cơ sở 3 này. Xin cảm ơn quý Thầy Cô bộ môn Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu trong năm học vừa qua.

Chúng em xin chân thành tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Anh Tuấn đã tận tình hướng dẫn em trong suốt quá trình thực hiện làm đồ án này.

Cảm ơn các bạn trong lớp, trong trường, trong ngành Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã hỗ trợ, giúp đỡ, chia sẻ kinh nghiệm và kiến thức cho nhóm trong quá trình thực hiện đề tài.

Trong quá trình làm đồ án, do kiến thức về lập trình di động chưa thực sự sâu sắc nên đồ án của chúng em không thể tránh khỏi những sai sót. Mong cô thông cảm và góp ý thêm cho chúng em để đồ án được hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Lý do chọn đề tài

Hiện nay, công nghệ thông tin được xem là một ngành mũi nhọn của các quốc gia, đặc biệt là các quốc gia đang phát triển, tiến hành công nghiệp hóa và hiện đại hoá như nước ta. Sự bùng nổ thông tin và sự phát triển mạnh mẽ của công nghệ kỹ thuật số, yêu cầu muốn phát triển thì phải tin học hoá vào tất cả các ngành các lĩnh vực.

Cùng với sự phát triển nhanh chóng về phần cứng máy tính, các ứng dụng ngày càng trở nên đa dạng, phong phú, hoàn thiện hơn và hỗ trợ hiệu quả cho con người. Các ứng dụng hiện nay ngày càng mô phỏng được rất nhiều nghiệp vụ khó khăn, hỗ trợ cho người dùng thuận tiện sử dụng, thời gian xử lý nhanh chóng, và một số nghiệp vụ được tự động hoá cao.

Do vậy mà trong việc phát triển ứng dụng, sự đòi hỏi không chỉ là sự chính xác, xử lý được nhiều nghiệp vụ thực tế mà còn phải đáp ứng các yêu cầu khác như về tốc độ, giao diện thân thiện, mô hình hoá được thực tế vào máy tính để người sử dụng tiện lợi, quen thuộc, tính tương thích cao, bảo mật cao (đối với các dữ liệu nhạy cảm), … Các ứng dụng giúp tiết kiệm một lượng lớn thời gian, công sức của con người, và tăng độ chính xác và hiệu quả trong công việc (nhất là việc sửa lỗi và tự động đồng bộ hoá).

Một ví dụ cụ thể, việc quản lý học vụ trong trường học. Nếu không có sự hỗ trợ của tin học, việc quản lý này phải cần khá nhiều người, chia thành nhiều khâu, mới có thể quản lý được toàn bộ hồ sơ, học sinh (thông tin, điểm số, kỷ luật, học bạ, …) lớp học (sỉ số, GVCN, thời khoá biểu, …), giáo viên (thông tin, lịch dạy, …) … cũng như các nghiệp vụ sắp thời khoá biểu, tính điểm trung bình, xếp loại học lực cho học sinh toàn trường (số lượng học sinh có thể lên đến hai ba ngàn học sinh). Các công việc này đòi hỏi nhiều thời gian và công sức, mà sự chính xác và hiệu quả không cao, vì đa số đều làm bằng thủ công rất ít tự động. Một số nghiệp vụ như tra cứu, thống kê, và hiệu chỉnh thông tin khá vất vả. Ngoài ra còn có một số khó khăn về vấn đề lưu trữ khá đồ sộ, dễ bị thất lạc, tốn kém, … Trong khi đó, các nghiệp vụ này hoàn toàn có thể tin học hoá một cách dễ dàng. Với sự giúp đỡ của tin học, việc quản lý học vụ sẽ trở nên đơn giản, thuận tiện, nhanh chóng và hiệu quả hơn rất nhiều.

## 1.2. Mục tiêu nghiên cứu

Xây dựng một ứng dụng hoàn thiện về giao diện và tối ưu về chức năng. Ứng dụng sẽ giúp người dùng dễ dàng truy cập và quản lý. Ứng dụng phát triên với một số chức năng cơ bản như thêm, sửa, xóa các danh mục như học sinh, điêm học sinh, thời khóa biểu, điểm danh.

- Xây dựng một hệ thống mới phù hợp, dể hiểu, dể sử dụng cho người dùng phần mềm. Hiểu được hệ thống quản lý học sinh, cần quản lý những mảng nào để xây dựng ứng dụng cho phù hợp nhu cầu thực tiễn.

- Giúp quản lý hệ thống của trường trở nên tối ưu hơn, công tác quản lý dễ dàng và tiện lợi hơn.

- Giúp bản thân có thêm khả năng sáng tạo, tư duy thông qua thiết kế giao diện tương tác với người dùng, vận dụng kiến thức bản thân vào thực tiễn.

## 1.3. Nhiệm vụ nghiên cứu

Phần mềm này được tạo ra nhằm đem đến sự thuận tiện cho các giáo viên cũng như giúp cho các trường học có thể xem xét một cách tổng quát tình hình học sinh, sinh viên của trường mình.

## 1.4. Phương pháp và phạm vi nghiên cứu

**Phương pháp nghiên cứu:**

- Tham khảo, nghiên cứu thêm các tài liệu trên trang web, trang báo, tìm hiểu tình hình thực trạng của đề tài.

- Tìm hiểu các đề tài của các anh chị khóa trước đã làm hoặc tìm hiểu qua sách báo, trang mạng liên quan đến đề tài nghiên cứu.

- Quan sát việc quản lý của một trường học trên địa bàn thành phố Đà Nẵng và nơi mình sinh sống, lấy thông tin thực tế từ giáo viên, phụ huynh, nhà trường... rồi tổng hợp, phân tích và thống kê số liệu.

**Phạm vi nghiên cứu:** Xoay quanh các trường học, lấy ý kiến từ phụ huynh học sinh

## 1.5. Phương tiện nghiên cứu

- Nghiên cứu cơ sở lý thuyết về phân tích và thiết kế hệ thống thông tin.

- Hệ quản trị CSDL và ngôn ngữ lập trình Java, PHP.

- Sử dụng các công cụ Android Studio, Sublime Text 3 để lập trình, thiết kế giao diện. - Sử dụng hệ quản trị cơ sở dữ liệu phpMyAdmin để quản trị cơ sở dữ liệu cho đề tài.

## 1.6. Xác định những yêu cầu hệ thống

### 1.6.1. Yêu cầu chức năng

Với một lượng lớn học sinh được tuyển vào từng năm câu hỏi đối với hệ thống mới: nhanh, tiện, chính xác để nhà trường dễ quản lý quá trình học của học sinh mà không có nhiều thời gian, sức mạnh.

Mới hệ thống phải rút ngắn được điểm nhập thời gian của giáo viên, đơn giản hóa quá điểm nhập.

Bao gồm hệ thống:

- Quản lý giáo viên thông tin

- Điểm quản lý.

- Lớp học thông tin quản lý.

- Duyệt duyệt tiêu chuẩn quản lý: rèn luyện hè, khen thưởng, tốt nghiệp.

Những người quản lý thông tin được cập nhật chính xác, những người thay đổi, những người có quyền lợi thì dễ dàng truy cập để theo dõi thông tin. Hệ thống hoạt động một cách tự động.

Thông tin hệ thống của hệ thống có thể được liên kết với dữ liệu excel. Biểu mẫu phần mềm được đưa ra cho luyện tập lưu trữ và tổng kết từng học kỳ, cả năm.

Phần mềm có thông báo ứng dụng đối với những kiến ​​nghị trường hợp, thông báo chế độ… (đối với những người có quyền truy cập).

Kết luận: Giải pháp cho hệ thống là lập trình quản lý trang web trên một nội bộ mạng hệ thống

### 1.6.2. Yêu cầu phi chức năng

Về thiết bị:

- Phần cứng: máy tính, mạng để phục vụ cho việc trao đổi dữ liệu và truy cập dễ dàng.

- Ứng dụng có thiết kế giao diện đẹp mà đơn giản, không rườm rà.

- Dễ sử dụng đối với người dùng.

- Bố cục các phần của ứng dụng rõ ràng.

## 1.7. Kết quả đề tài

Hoàn thiện ứng dụng về mặt giao diện và một số chức năng cơ bản.Ứng dụng hoàn thiện tạo nên một mối liên hệ mới giữa giáo viên và và việc quản lý học sinh từ đó dễ dàng hơn trong việc theo dõi tình hình học tập của từng cá nhân học sinh. Ứng dụng với một số chức năng chính sau dành cho phân quyền giáo viên:

- Quản lý điểm danh, điểm học sinh, nhận xét của giáo viên đến từng phụ huynh

- Quản lý thời khóa biểu của học sinh, lịch báo giảng của giáo viên

- Quản lý thông tin của từng học sinh

Sau khi kết thúc một dự án thứ không thể thiếu chính là báo cáo viết về đề tài của mình và slide trình bày dự án. Báo cáo chi tiết trình bày bố cục mục lục hợp lý.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Công cụ thiết kế ứng dụng

### 2.1.1. Android studio

Android Studio là [môi trường chính thức  phát triển tích hợp](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) cho [Google](https://en.wikipedia.org/wiki/Google)  hệ điều hành Android, xây dựng trên [JetBrains](https://en.wikipedia.org/wiki/JetBrains) ' [IDEA IntelliJ](https://en.wikipedia.org/wiki/IntelliJ_IDEA) phần mềm và thiết kế đặc biệt cho [phát triển Android](https://en.wikipedia.org/wiki/Android_software_development). Nó có sẵn để tải xuống trên các hệ điều hành dựa trên [Windows](https://en.wikipedia.org/wiki/Windows) , [macOS](https://en.wikipedia.org/wiki/MacOS) và [Linux](https://en.wikipedia.org/wiki/Linux) hoặc dưới dạng dịch vụ dựa trên đăng ký vào năm 2020.  Nó là sự thay thế cho [công cụ phát triển Android Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (E-ADT) như là IDE chính để phát triển ứng dụng Android gốc.

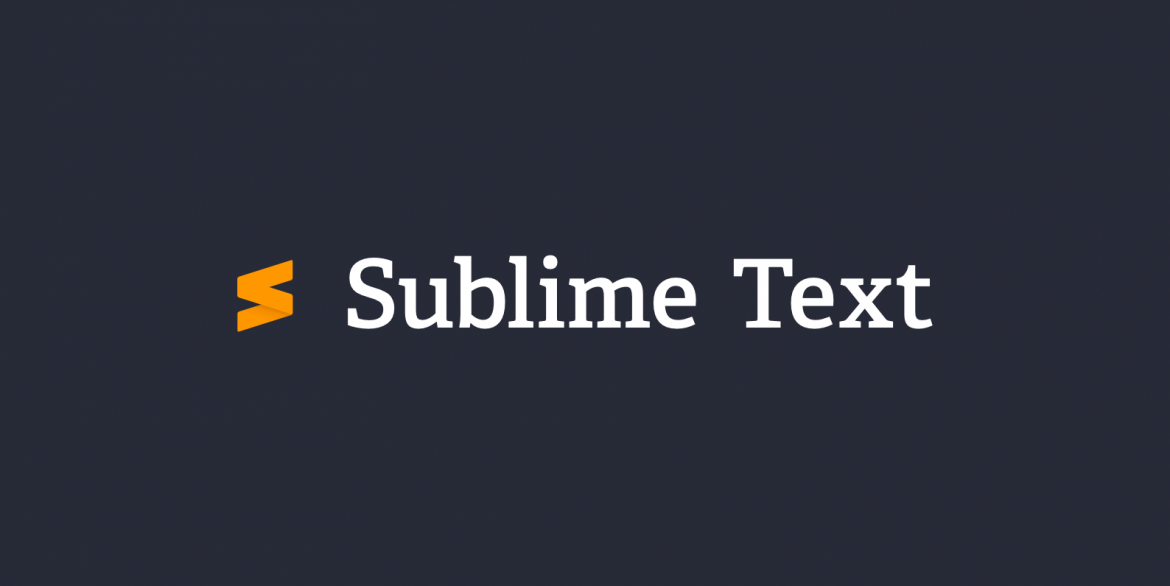


***Hình 2 - 1: Công cụ lập trình Android Studio***

Android Studio được công bố vào ngày 16 tháng 5 năm 2013 tại hội nghị [Google I / O.](https://en.wikipedia.org/wiki/Google_I/O)Nó đang ở giai đoạn xem trước truy cập sớm bắt đầu từ phiên bản 0.1 vào tháng 5 năm 2013, sau đó bước vào giai đoạn beta bắt đầu từ phiên bản 0.8 được phát hành vào tháng 6 năm 2014.

### 2.1.2. Sublime Text 3

Là công cụ được sử dụng trong việc viết code api cho framework Laravel. Sublime Text 3 (ST3) là phiên bản mới nhất của chương trình soạn thảo văn bản được sử dụng nhiều nhất bởi web developers, coders và programmers. Nó sẵn có cho Mac,

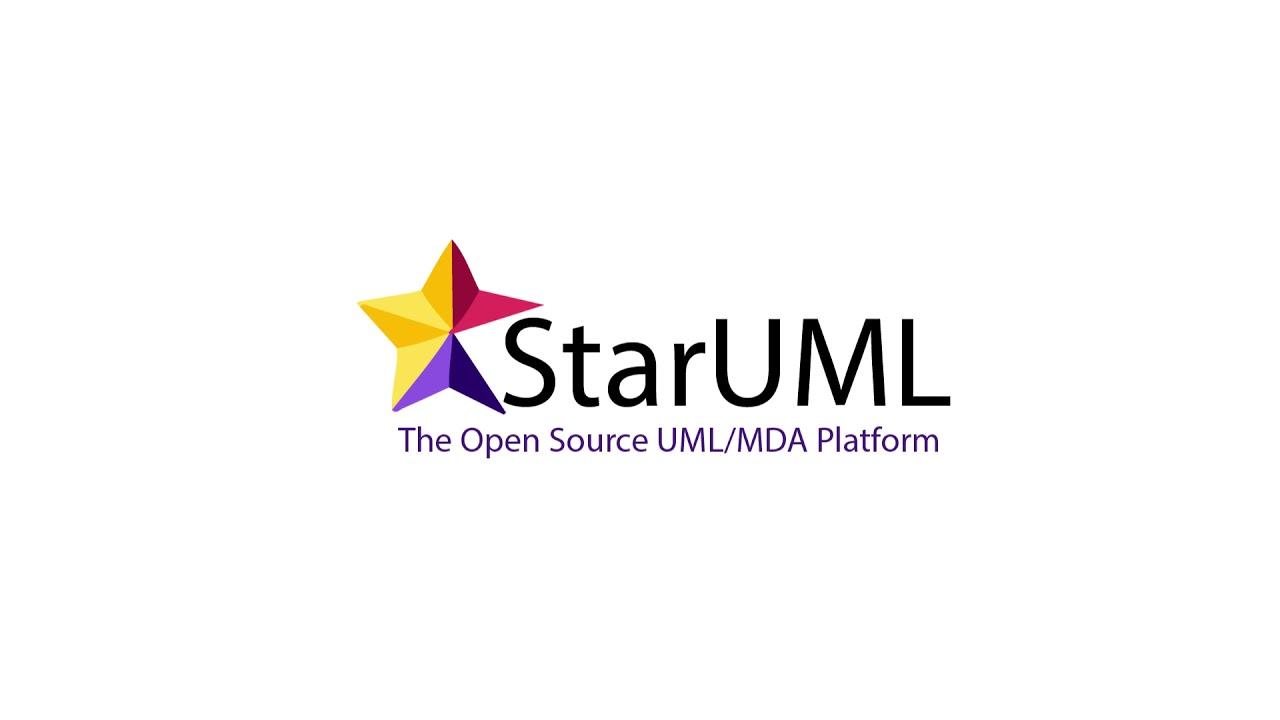


***Hình 2 - 2: Công cụ lập trình Sublime Text 3***

Windows, Linux và cho phép tải về và sử dụng miễn phí. Trong bài viết này chúng ta sẽ cùng nhau tìm hiểu về hướng dẫn sử dụng Sublime Text 3 cho web developer, phím tắt hữu ích và hơn thế nữa.

### 2.1.3. Phần mềm phân tích thiết kế hệ thống StarUML

StarUML là một mô hình nền tảng, là phần mềm hỗ trợ UML (Unified Modeling Language). Nó hỗ trợ các phương pháp tiếp cận MDA (Model Driven Architecture) bằng cách hỗ trợ các khái niệm hồ sơ UML. Tức là StarUML hỗ trợ phân tích và thiết kế hệ thống một điều mà bất cứ dự án nào đều cần có. Ngoài ra dùng StarUML sẽ đảm bảo tối đa hóa năng suất và chất lượng của các dự án phần mềm của bạn. Vì nó cho phép mô hình hóa nên sẽ không phụ thuộc vào người code, ngôn ngữ code hay nền tảng sử dụng. Do mô hình hóa nên rất dễ đọc và dễ hiểu. Một điều mình thích nhất ở StarUML là khả năng tự sinh code từ những mô hình class.



***Hình 2 - 3: Phần mềm phân tích thiết kế hệ thống StarUML***

**Tự sinh code bằng StarUML**

Không phụ thuộc vào nền tảng, ngôn ngữ nên StarUML có thể biên dịch ra tất cả các ngôn ngữ (mới thử với PHP). Tức là từ các Diagram ta có thể sinh ra code của một ngôn ngữ một cách tự động. Thật tuyệt khi không cần code chỉ cần thiết kế là code tự sinh! Sau đây mình xin giới thiệu tính năng tuyệt vời này.

### 2.1.4. Công cụ quản trị cơ sở dữ liệu PhpMyAdmin

PhpMyAdmin là phần mềm mã nguồn mở được viết bằng ngôn ngữ PHP giúp quản trị cở sở dữ liệu MySQL thông qua giao diện web**.** Tính đến nay, phpMyAdmin đã có đến hàng triệu lượt sử dụng và vẫn không ngừng tăng.

- MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.

## 2.2. Ngôn ngữ thiết kế ứng dụng

### 2.2.1. Ngôn ngữ chính lập trình ứng dụng android

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem. Java được tạo ra với tiêu chí “*Viết (code) một lần, thực thi khắp nơi*” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

**Đặc điểm**

Tương tự C++, hướng đối tượng hoàn toàn. Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++. Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch**.** Cơ chế thu gom rác tự động: Khi tạo ra các đối tượng trong Java, JRE sẽ tự động cấp phát không gian bộ nhớ cho các đối tượng ở trên heap.

Đa luồng: Java hỗ trợ lập trình đa tiến trình (multithread) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority…).



***Hình 2 - 4: Ngôn ngữ lập trình Java***

### 2.2.2. Ngôn ngữ XML (eXtensible Markup Language)

XML (viết tắt từ [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *eXtensible Markup Language*, tức "Ngôn ngữ đánh dấu mở rộng") là [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) với mục đích chung do [W3C](https://vi.wikipedia.org/wiki/W3C) đề nghị, để tạo ra các ngôn ngữ đánh dấu khác. Đây là một tập con đơn giản của [SGML](https://vi.wikipedia.org/wiki/SGML), có khả năng mô tả nhiều loại [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) khác nhau. Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với [Internet](https://vi.wikipedia.org/wiki/Internet). Các ngôn ngữ dựa trên XML (Ví dụ: [RDF](https://vi.wikipedia.org/wiki/RDF), [RSS](https://vi.wikipedia.org/wiki/RSS_(%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng_t%E1%BA%ADp_tin)), [MathML](https://vi.wikipedia.org/wiki/MathML), [XHTML](https://vi.wikipedia.org/wiki/XHTML), [SVG](https://vi.wikipedia.org/wiki/SVG), [GML](https://vi.wikipedia.org/wiki/GML) và [cXML](https://vi.wikipedia.org/wiki/CXML)) được định nghĩa theo cách thông thường, cho phép các chương trình sửa đổi và kiểm tra hợp lệ bằng các ngôn ngữ này mà không cần có hiểu biết trước về hình thức của chúng.



***Hình 2 - 5: Ngôn ngữ XML(eXtensible Markup Language)***

**So sánh XML với HTML**

XML và HTML giống nhau đều là các thẻ (tag)

Tuy nhiên XML không phải là sự thay thế cho HTML.

| **XML** | **HTML** |
| --- | --- |
| XML cho phép người dùng tạo biểu tượng đánh dấu riêng để mô tả nội dung, tạo một biểu tượng không giới hạn và tự định nghĩa | Được định nghĩa trước và người dùng phải tuân thủ |
| Được thiết kế để chuyển tải và lưu trữ dữ liệu | Được thiết kế để hiển thị dữ liệu |

***Bảng 2 - 1: So sánh XML với HTML***

### 2.2.3. Ngôn ngữ thiết kế hệ thống UML

UML (Unified Modeling Language) là ngôn ngữ dành cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm.

UML tạo cơ hội để viết thiết kế hệ thống, bao gồm những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống.Cách xây dựng các mô hình trong UML phù hợp mô tả các hệ thống thông tin cả về cấu trúc cũng như hoạt động. Cách tiếp cận theo mô hình của UML giúp ích rất nhiều cho những người thiết kế và thực hiện hệ thống thông tin cũng như những người sử dụng nó; tạo nên một cái nhìn bao quát và đầy đủ về hệ thống thông tin dự định xây dựng. Cách nhìn bao quát này giúp nắm bắt trọn vẹn các yêu cầu của người dùng; phục vụ từ giai đoạn phân tích đến việc thiết kế, thẩm định và kiểm tra sản phẩm ứng dụng công nghệ thông tin.



***Hình 2 - 6: Ngôn ngữ thiết kế hệ thống UML***

UML sử dụng một hệ thống ký hiệu thống nhất biểu diễn các Phần tử mô hình (model elements). Tập hợp các phần tử mô hình tạo thành các Sơ đồ UML (UML diagrams). Có các loại sơ đồ UML chủ yếu sau:

- Sơ đ**ồ** lớp (Class Diagram)

- Sơ đồ đối tượng (Object Diagram)

- Sơ đồ tình huống sử dụng (Use Cases Diagram)

- Sơ đồ trình tự (Sequence Diagram)

- Sơ đồ cộng tác (Collaboration Diagram hay là Composite Structure Diagram)

- Sơ đồ trạng thái (State Machine Diagram)

- Sơ đồ thành phần (Component Diagram)

- Sơ đồ hoạt động (Activity Diagram)

- Sơ đồ triển khai (Deployment Diagram)

- Sơ đồ gói (Package Diagram)

- Sơ đồ liên lạc (Communication Diagram)

- Sơ đồ tương tác (Interaction Overview Diagram - UML 2.0)

- Sơ đồ phối hợp thời gian (Timing Diagram - UML 2.0)

### 2.2.4. Ngôn ngữ lập trình PHP

PHP là viết tắt của từ Hypertext Preprocessor. Ngôn ngữ này được phát triển từ năm 1994 và cho đến nay đã được nhiều người sử dụng để phát triển các ứng dụng phần mềm thông qua lập trình web. Đặc điểm của ngôn ngữ lập trình này là sử dụng mã nguồn mở, dễ dàng nhúng vào HTML và tích hợp với web.



***Hình 2 - 7: Ngôn ngữ lập trình PHP***

Ưu điểm của ngôn ngữ lập trình PHP là cú pháp đơn giản, tốc độ xử lý nhanh, tính cộng đồng cao. Chính vì vậy mà lập trình PHP được sử dụng chủ yếu để thiết kế web. Để tạo ra các ứng dụng web bằng ngôn ngữ PHP, người lập trình phải sử dụng các dòng lệnh cơ bản.

Bạn có thể sử dụng ngôn ngữ lập trình PHP để tạo ra các blog cá nhân hoặc những website lớn. Bởi, ngôn ngữ này có nhiều Framework được xây dựng từ các mã nguồn mở như Wordpress, OpenCart.

**Tính năng chính của PHP**

- Mã nguồn mở

- Nền tảng độc lập

- Sử dụng quy trình và hướng đến đối tượng

- Dễ học

- Phát triển ứng dụng và mục đích sử dụng

- Tương thích với máy chủ

- Tích hợp cơ sở dữ liệu dễ dàng

### 2.2.2. **Framework Laravel**

Laravel là một PHP framework, có mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC.

Những lý do khiến Laravel trở nên rộng rãi:

- Cú pháp dễ hiểu – rõ ràng

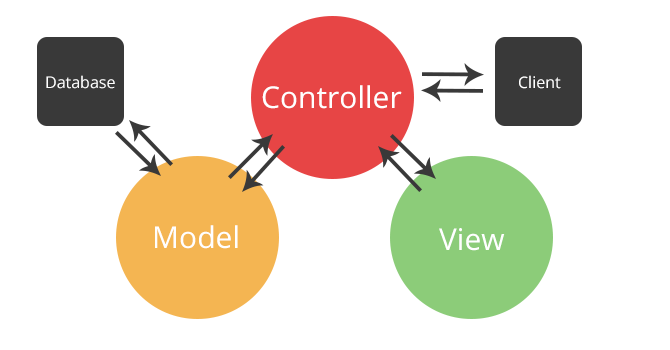
- Hệ thống đóng gói modular và quản lý gói phụ thuộc

- Nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ

- Nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

**Mô hình MVC**

MVC (Model-View-Controller) là mẫu kiến trúc phần mềm trên máy tính nhằm mục đích tạo lập giao diện cho người dùng. Theo đó, hệ thống MVC được chia thành ba phần có khả năng tương tác với nhau và tách biệt các nguyên tắc nghiệp vụ với giao diện người dùng.



***Hình 2 - 8: Cấu trúc mô hình MVC***

Ba thánh phần ấy bao gồm:

- Controller: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng.

- Model: Là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý…

- View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images…

Bằng cách này, thông tin nội hàm được xử lý tách biệt với phần thông tin xuất hiện trong giao diện người dùng. Bảo đảm các nguyên tắc nghề nghiệp của lập trình viên.

## 2.3. Công nghệ xây dựng ứng dụng

### 2.3.1. Khái niệm về API RESTful API trong Moblie App

**API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.

**Web API hoạt động như thế nào?**

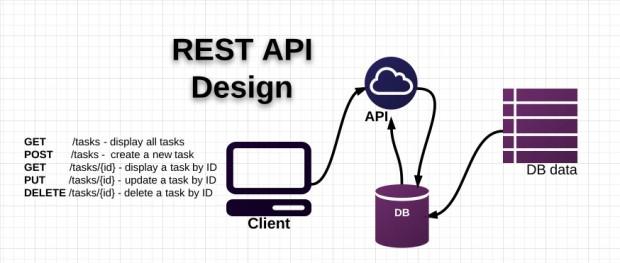
- Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

- Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

- Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

- Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động , dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu…

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP. Với sự gia tăng của phát triển phần mềm Mobile và JavaScript, sử dụng API RESTful là lựa chọn tốt nhất để xây dựng một giao diện duy nhất giữa dữ liệu của chúng ta và khách hàng.



***Hình 2 - 9: Mô hình RESTful API***

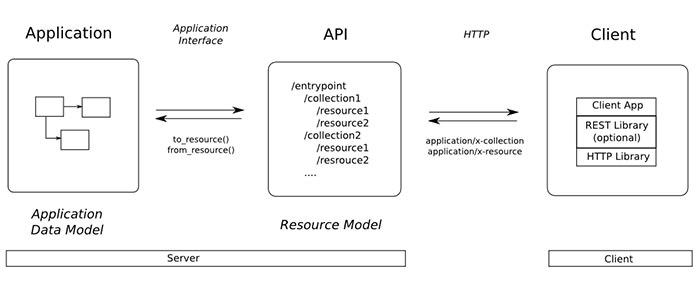
Trước tiên, chúng ta cần phải hiểu chính xác những gì được coi là một RESTful API. REST là viết tắt của *Reppresentational State Transfer* và được dùng để giao tiếp mạng giữa các ứng dụng dựa trên giao thức không trạng thái (thường là HTTP). Các dạng HTTP thường dùng:

- GET : truy xuất dữ liệu

- POST : tạo mới dữ liệu

- PUT : cập nhật dữ liệu

- DELETE : xóa dữ liệu



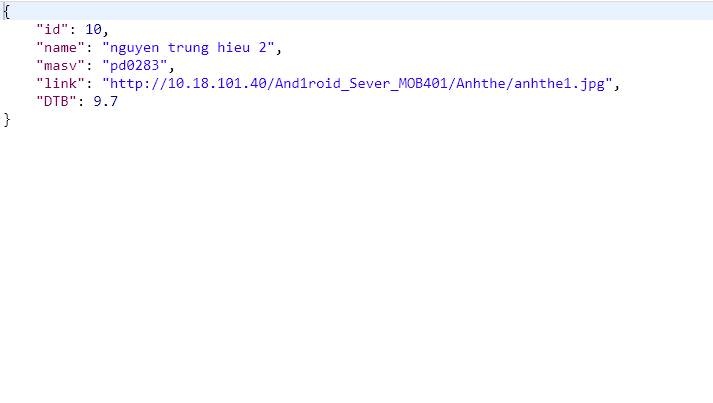
***Hình 2 - 10: Mô hình API chi tiết***

### 2.3.2. Khái niệm và cấu trúc json

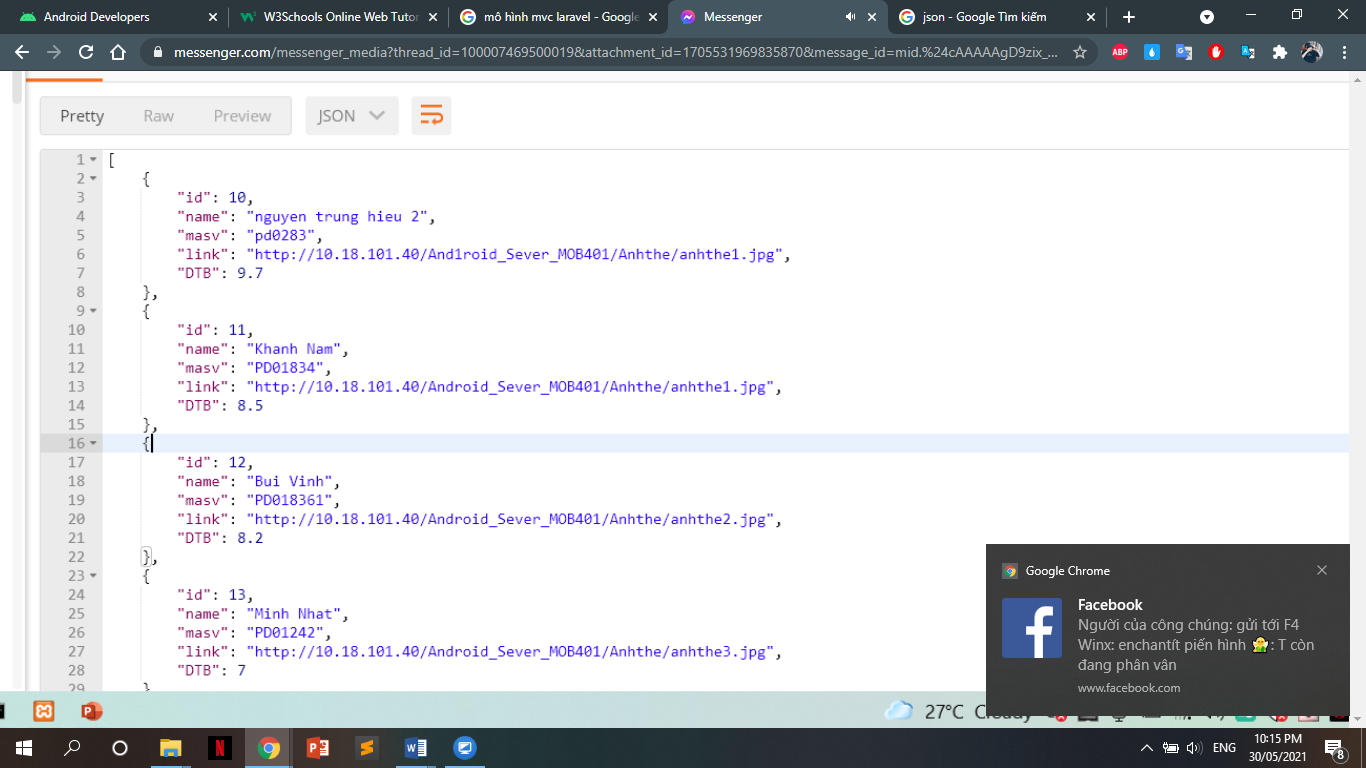
**JSON** là chữ viết tắt của **J**avascript **O**bject **N**otation, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được. JSON được coi là một tiêu chuẩn mở để có thể trao đổi những thông tin dữ liệu trên web.

**Dữ liệu từ web API trả về JSON**

- JSON học sinh:



- JSON danh sách lớp 1:



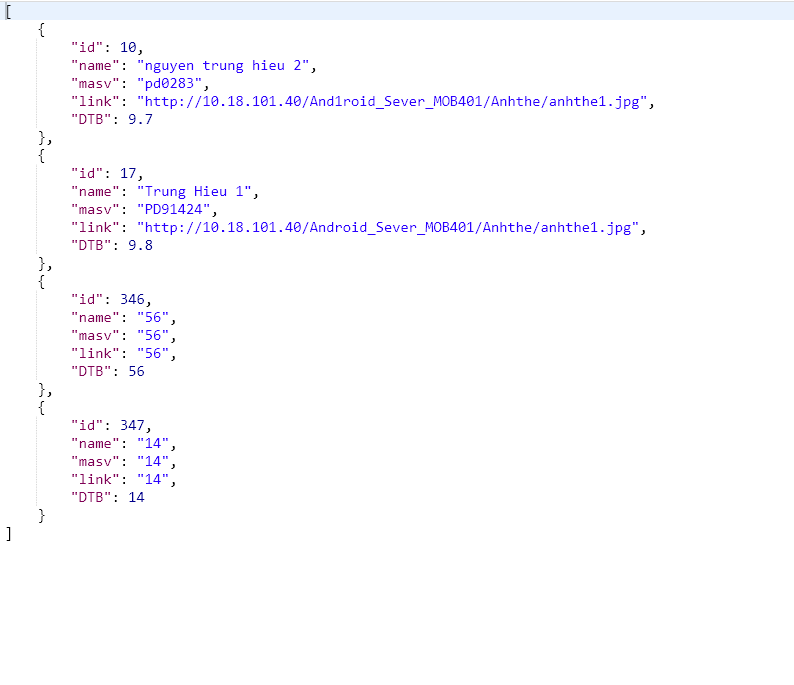
- JSON danh sách môn học:



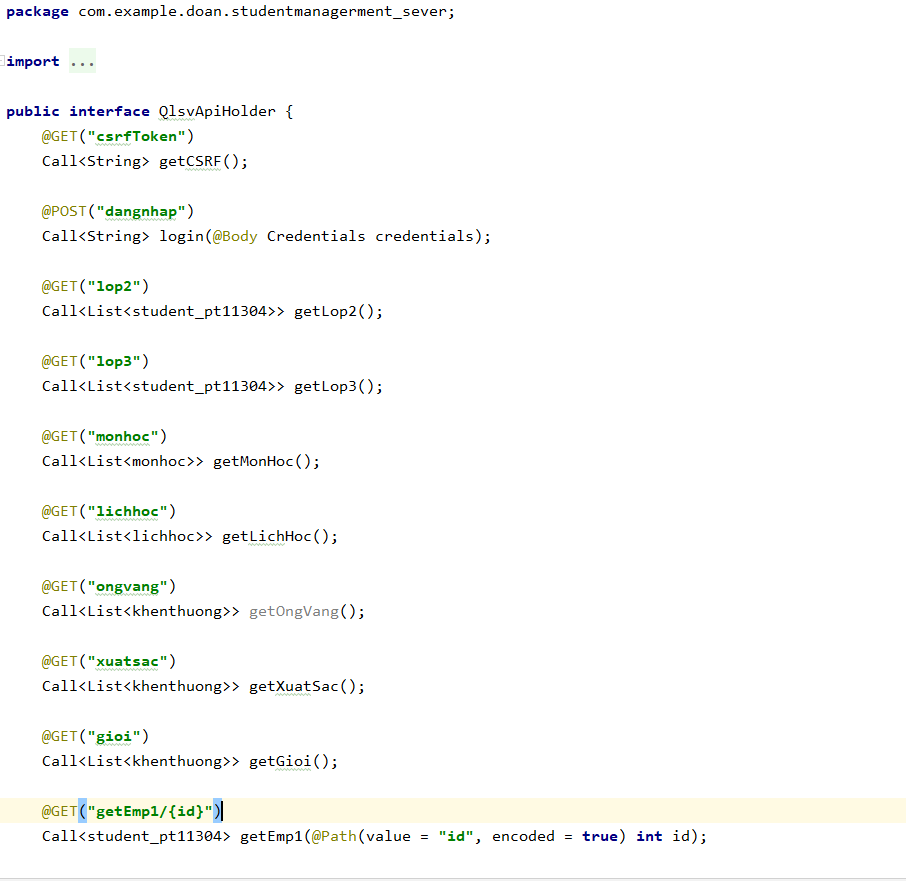
- JSON lịch học:

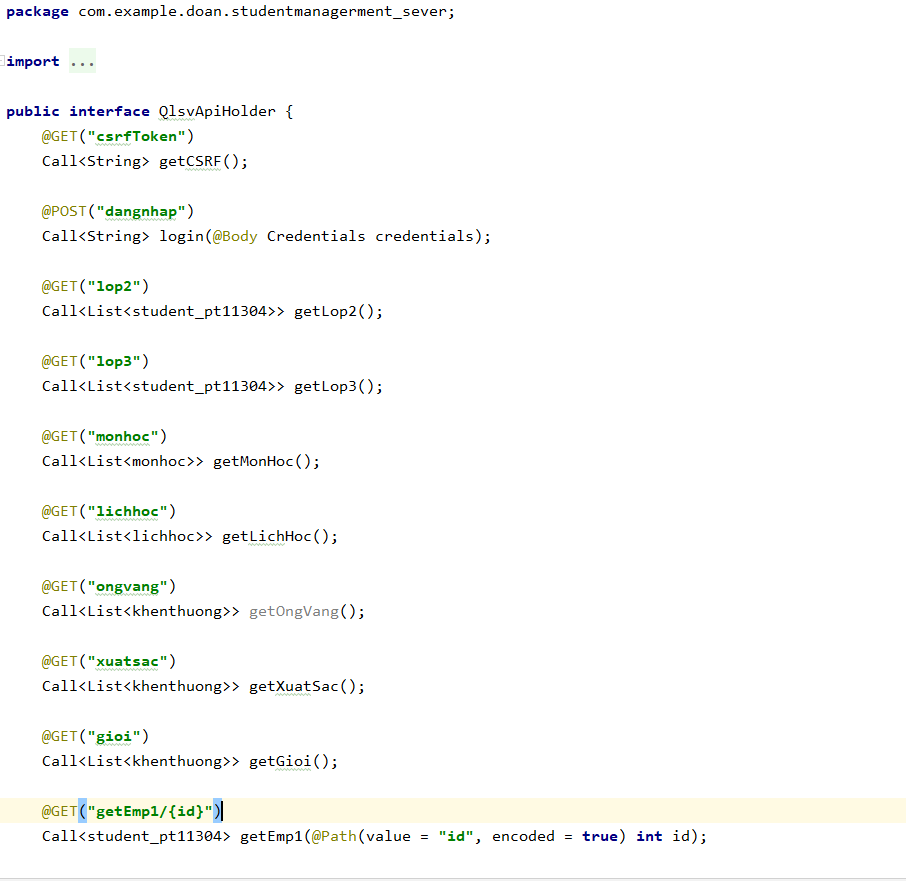


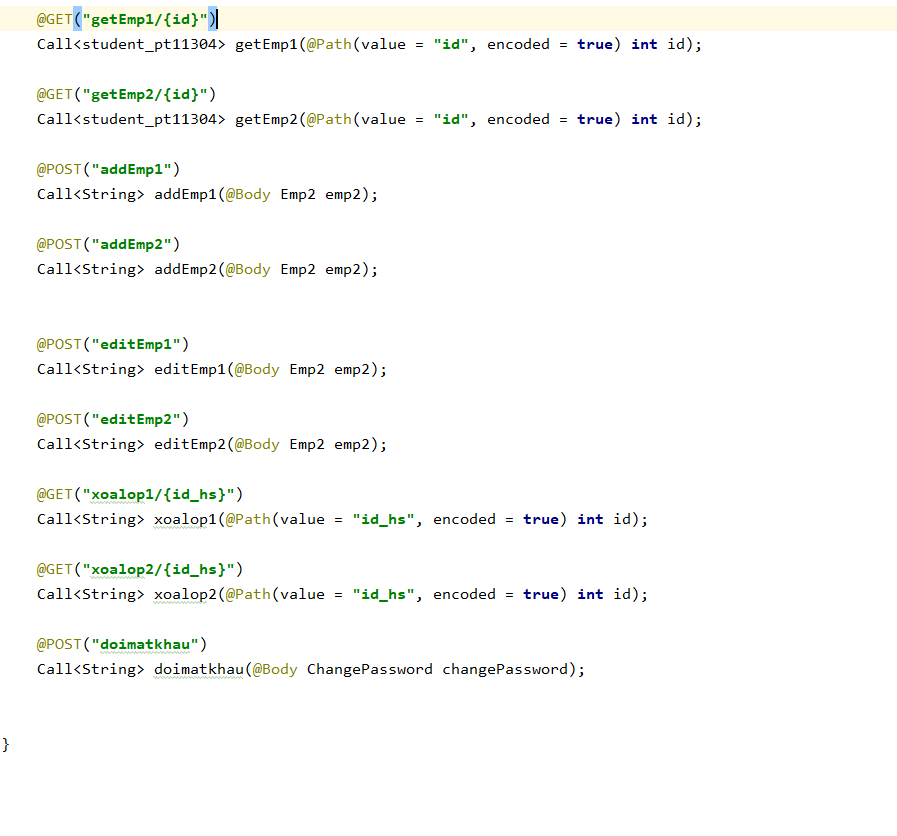
- JSON danh sách khen thưởng:



**Ứng dụng lấy dữ liệu từ JSON trả về thông qua giao thức không trạng thái HTTP:**



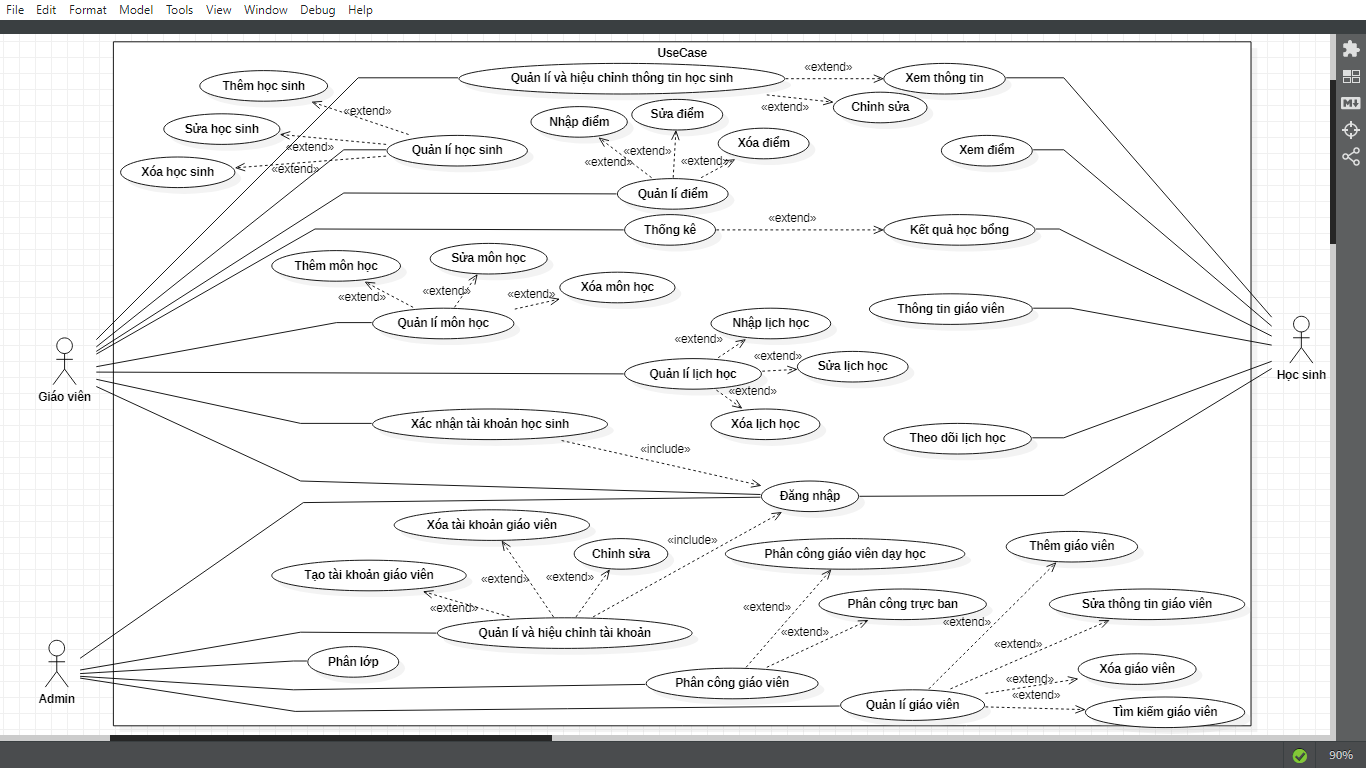




# CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

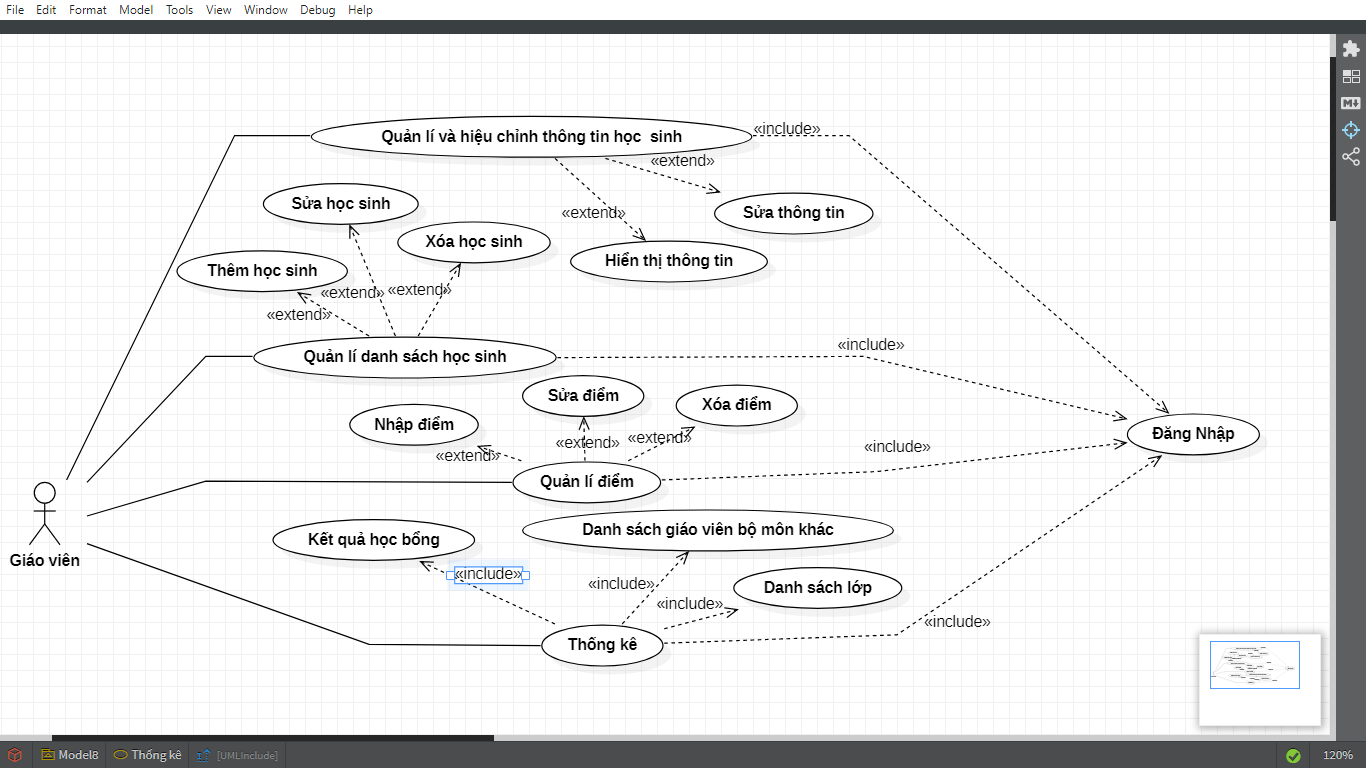
## 3.1. Biểu đồ Usecase

### 3.1.1. Biểu đồ Usecase tổng quát



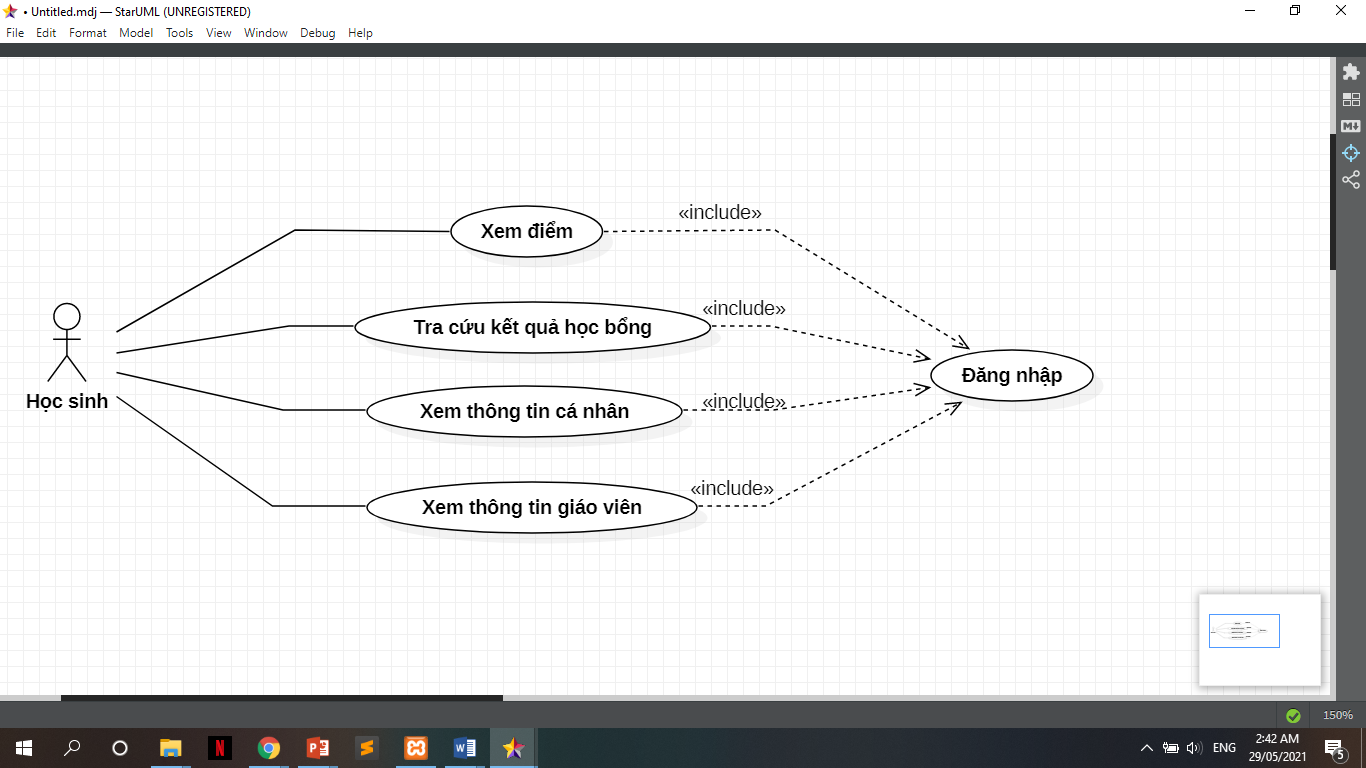
***Hình 3 - 1: Biểu đồ Use-case tổng quát***

### 3.1.2. Biểu đồ Usecase phân rã actor “Giáo Viên”



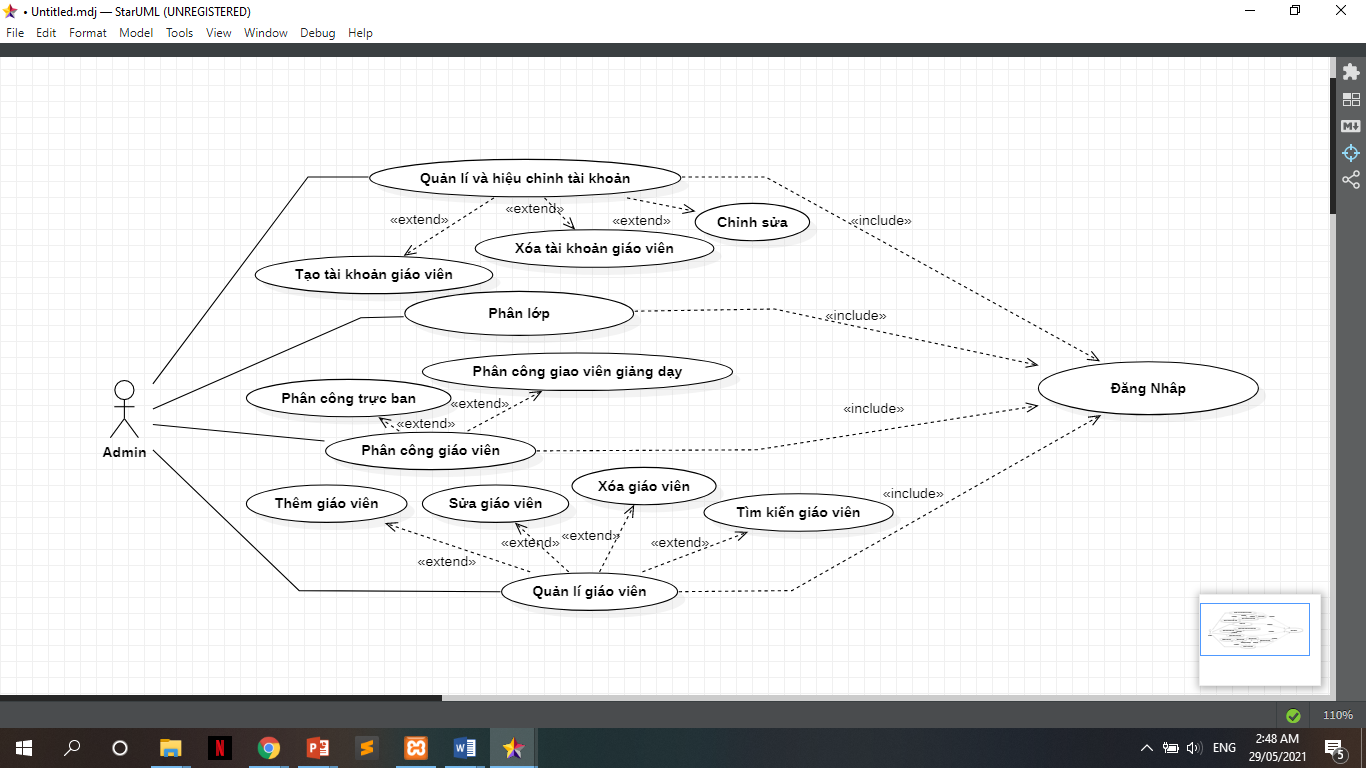
***Hình 3 - 2: Biểu đồ Usecase phân rã actor “Giáo Viên”***

### 3.1.3. Biểu đồ Usecase phân rã actor “Học Sinh”



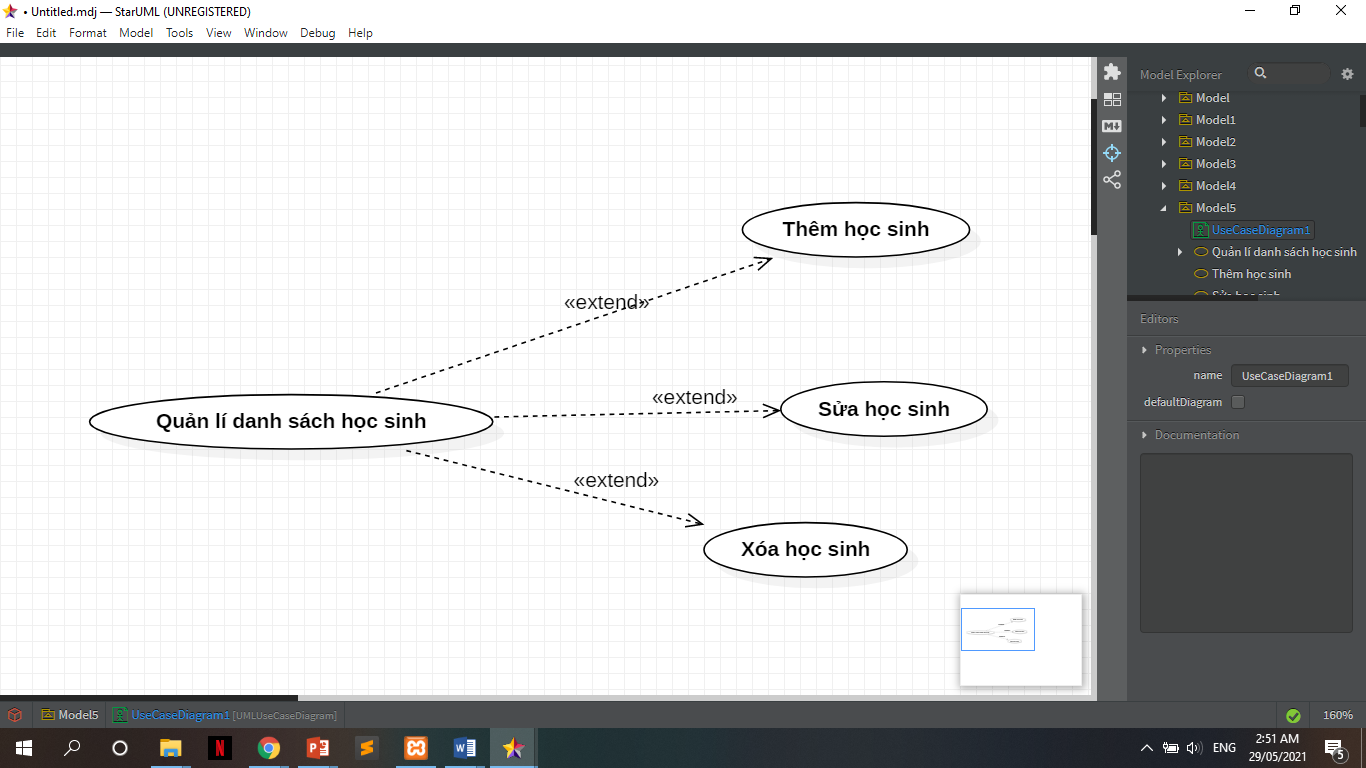
***Hình 3 - 3: Biểu đồ Usecase phân rã actor “Học Sinh”***

### 3.1.4. Biểu đồ Usecase phân rã actor “Admin”



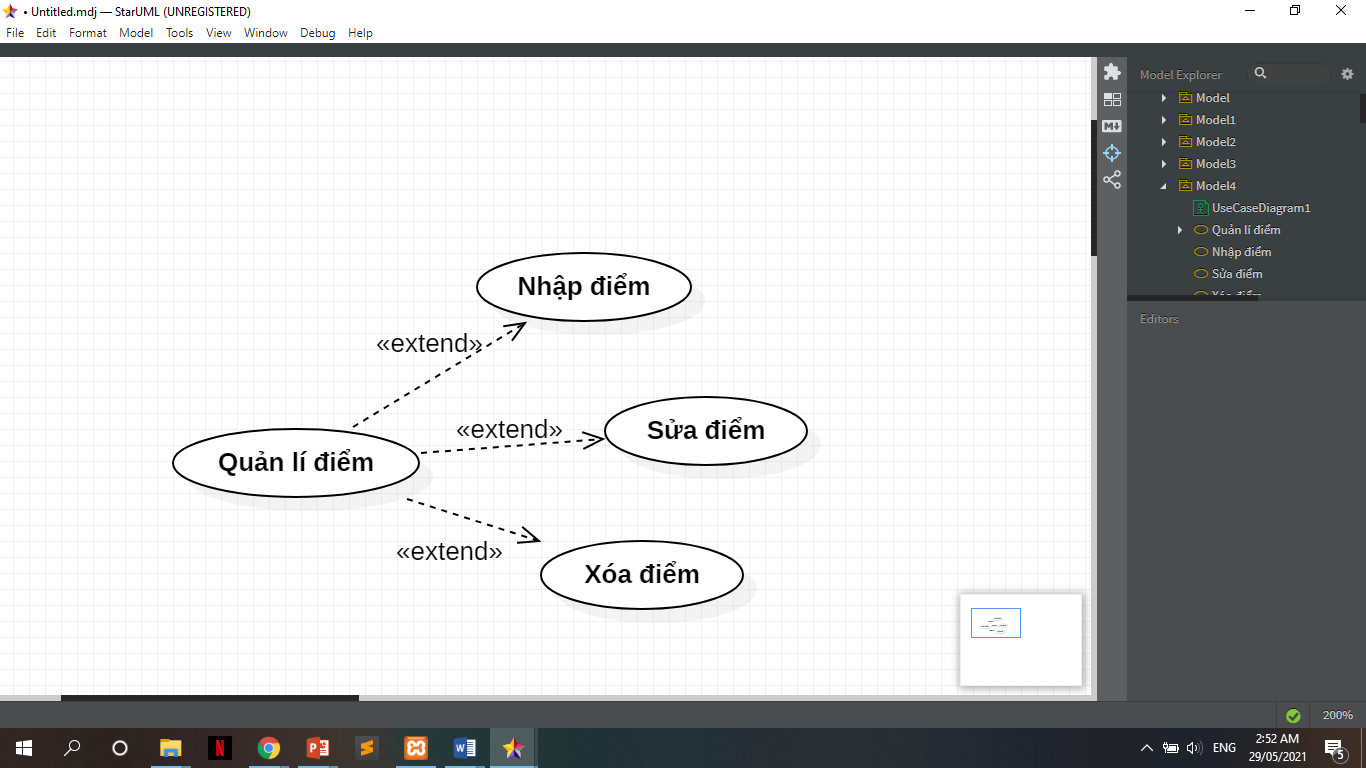
***Hình 3 - 4: Biểu đồ Usecase phân rã actor “Admin”***

### 3.1.5. Usecase quản lí danh sách học sinh



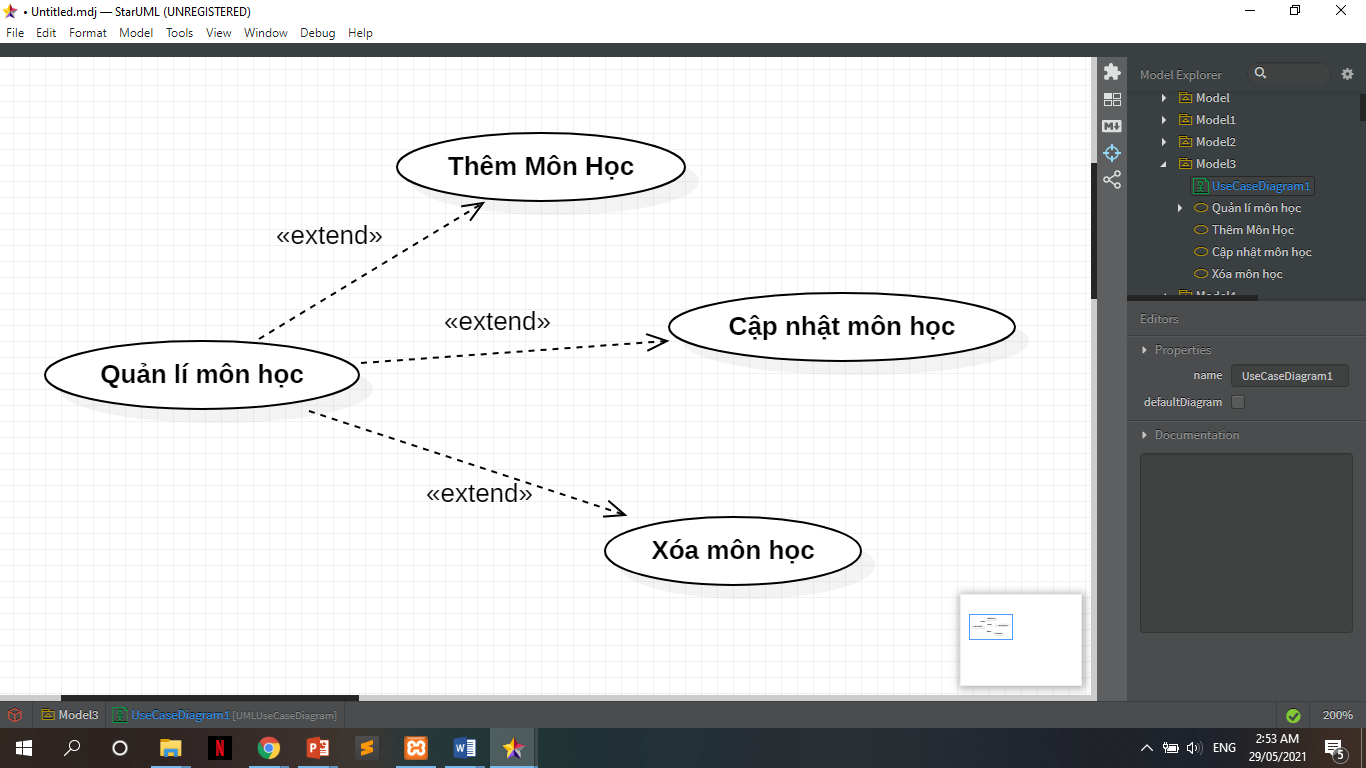
***Hình 3 - 5: Usecase quản lý học sinh***

### 3.1.6. Usecase quản lí điểm



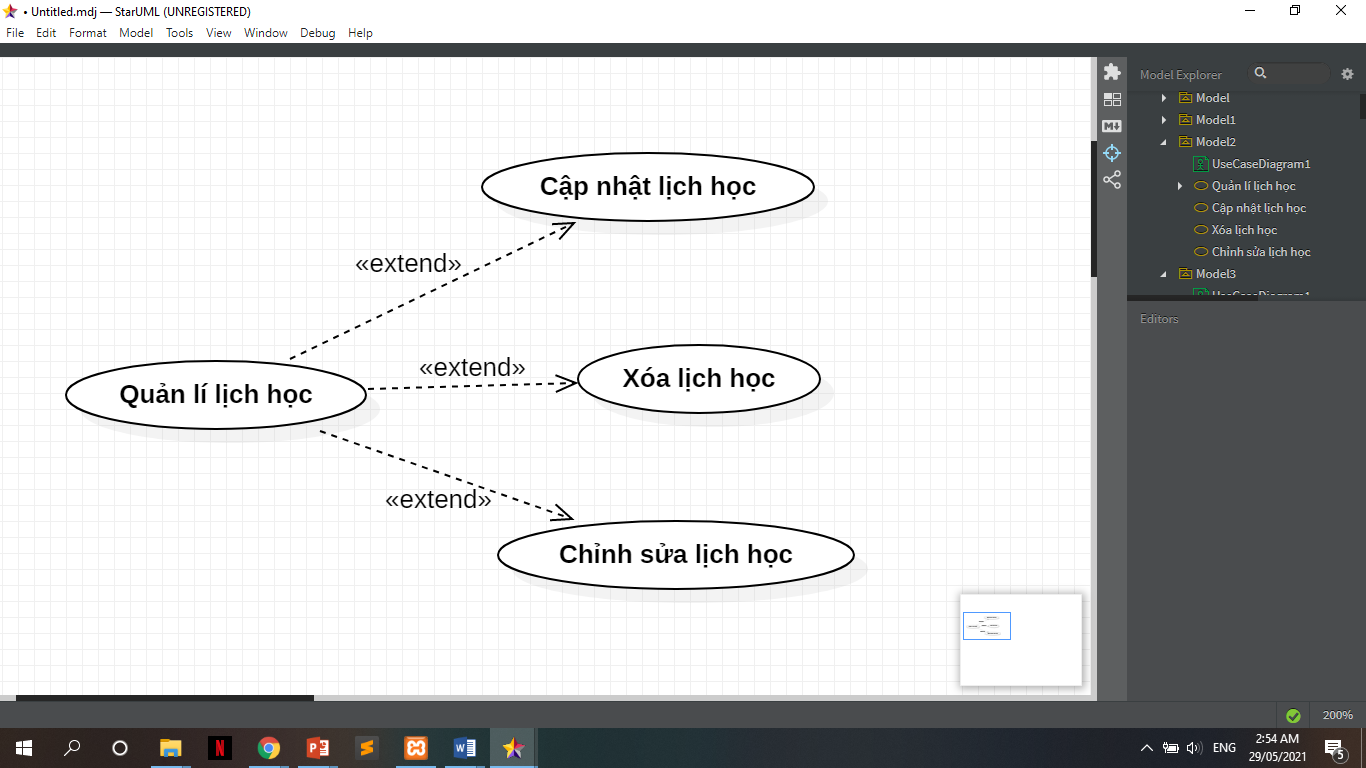
***Hình 3 - 6: Usecase quản lí điểm***

### 3.1.7. Usecase quản lí môn học



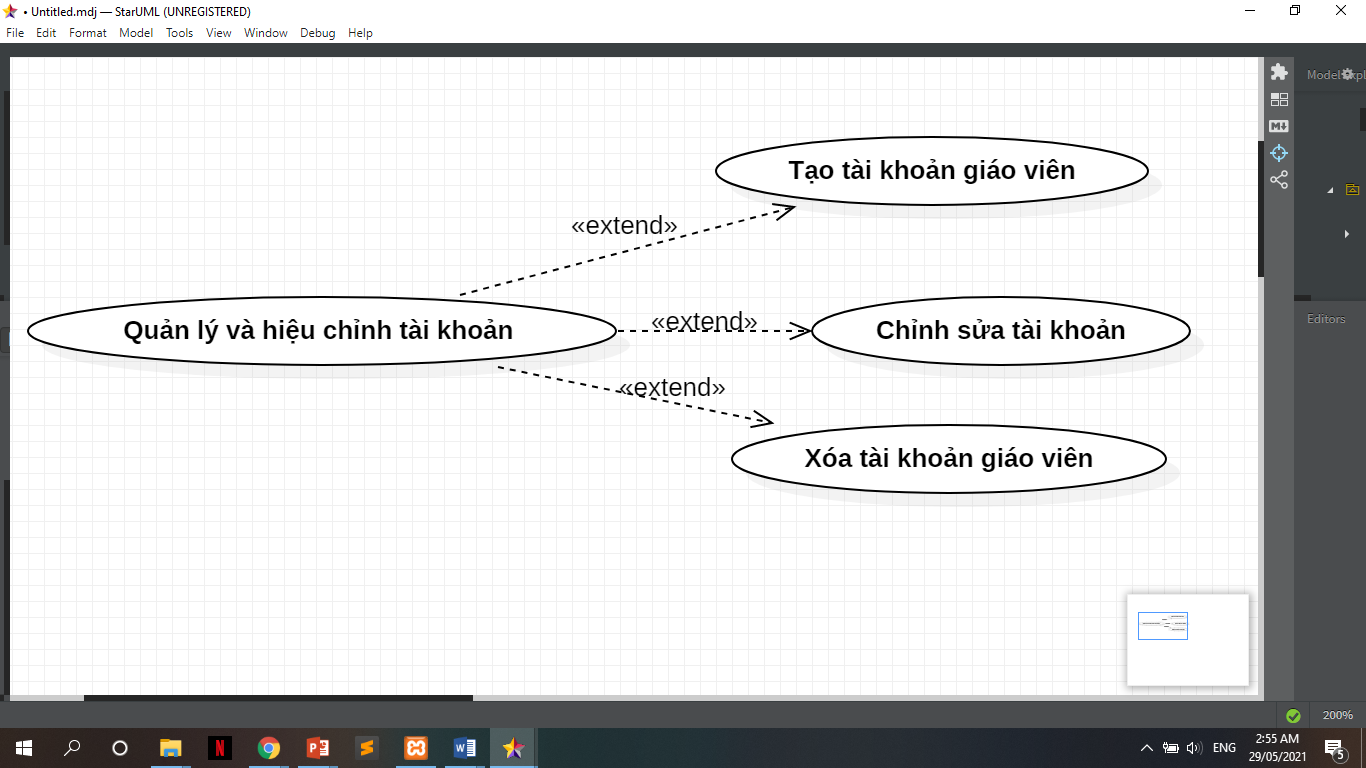
***Hình 3 - 7: Usecase quản lí môn học***

### 3.1.8. Usecase quản lí lịch học



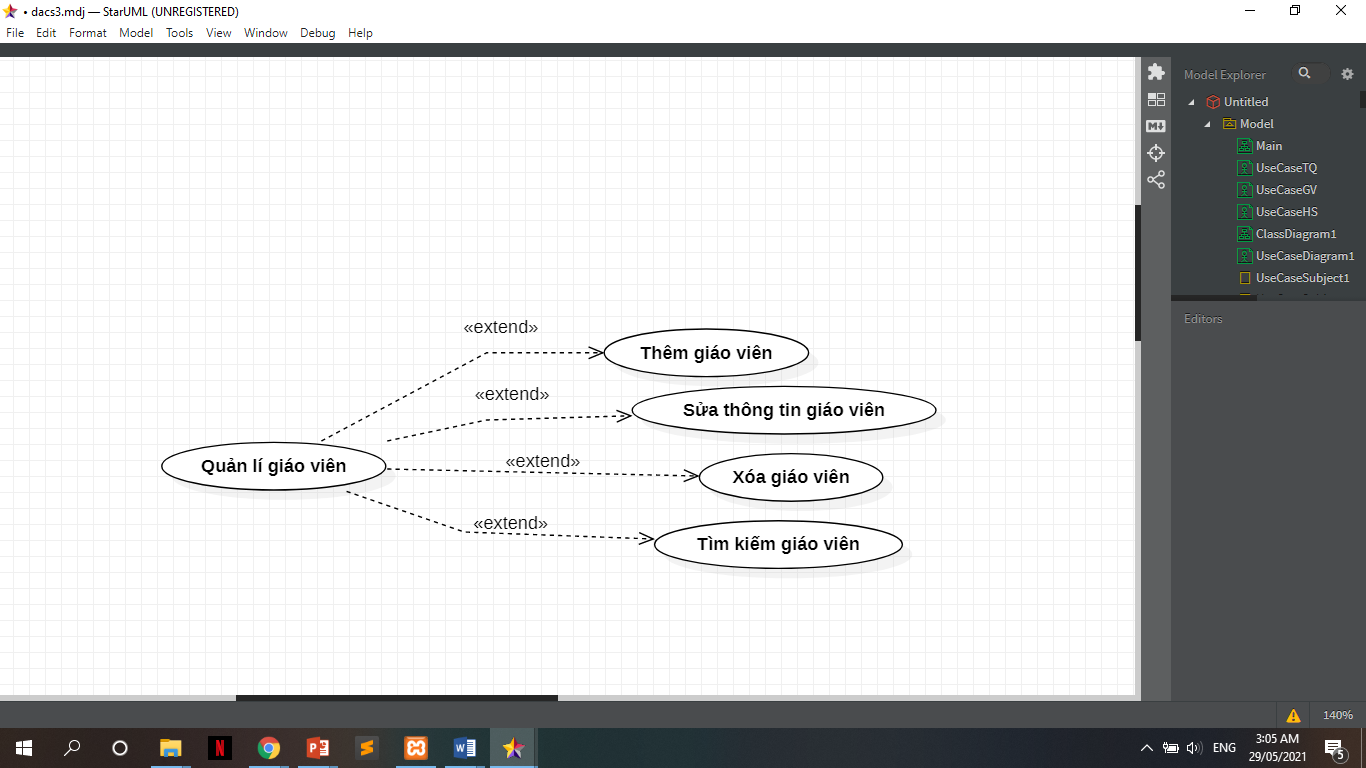
***Hình 3 - 8: Usecase quản lí lịch học***

### 3.1.9. Usecase quản lí và hiệu chỉnh tài khoản.



***Hình 3 - 9: Usecase quản lí và hiệu chỉnh tài khoản.***

### 3.1.10. Usecase quản lí giáo viên.



***Hình 3 - 10: Usecase quản lí giáo viên.***

## 3.2. Đặc tả Usecase

### 3.2.1. Đặc Usecase đăng nhập hệ thống

a) Mô tả tóm tắt

- Tên ca sử dụng: Đăng nhập hệ thống - Mục đích: Mô tả cách một người sử dụng đăng nhập vào hệ thống.

- Tác nhân: admin, giáo viên, học sinh (hoặc phụ huynh học sinh)

b) Các luồng sự kiện

**Các luồng sự kiện chính**

- Ca sử dụng này bắt đầu khi tác nhân muốn đăng nhập vào hệ thống.

- Hệ thống yêu cầu tác nhân nhập tài khoản và mật khẩu đăng nhập

- Tác nhân nhập tên đăng nhập và mật khẩu đăng nhập của mình.

- Hệ thống xác nhận tài khoản và mật khẩu đăng nhập có hợp lệ không, nếu không hợp lệ thì thực hiện Luồng A1.

- Hệ thống ghi lại quá trình đăng nhập.

**Các luồng rẽ nhánh**

- Luồng A1: Nhập sai tài khoản mật khẩu đăng nhập

+ Hệ thống hiển thị một thông báo lỗi.

+ Người sử dụng có thể chọn hoặc là đăng nhập lại hoặc là huỷ bỏ đăng nhập, khi đó ca sử dụng kết thúc.

c) Tiền điều kiện*:* Không

d) Hậu điều kiện: Nếu việc đăng nhập thành công, người sử dụng sẽ đăng nhập được vào hệ thống

### 3.2.2. Đặc tả Usecase quản lí và hiệu chỉnh thông tin học sinh

a) Mô tả tóm tắt

- Tên UC: Quản lý và hiệu chỉnh thông tin học sinh

- Mục đích: Mô tả người quản lý thêm, sửa, xóa thông tin học sinh

- Tác nhân: Giáo viên.

b) Các luồng sự kiện

**Luồng sự kiện chính:**

- Từ màn hình làm việc chính chọn chức năng Quản lý thông tin học sinh

- Hệ thống hiện thị các lựa chọn

+ Thêm

+ Sửa

- Thêm học sinh:

+ Người dùng muốn thêm học sinh mới vào trong hệ thống.

+ Người dùng chọn nút “Thêm mới”.

+ Hệ thống yêu cầu nhập thông tin cần thiết.

+ Người dùng nhập thông tin cần thiết vào trong hệ thống.

+ Người dùng chọn nút “Lưu” (Luồng A1).

+ Hệ thống kiểm tra dữ liệu nhập vào (Luồng A3).

+ Thêm thành công, kết thúc use case.

+ Kết quả: Một học sinh mới được thêm vào.

- Sửa thông tin học sinh.

+ Người dùng chọn học sinh cần sửa thông tin.

+ Người dùng nhập thông tin cần thay đổi vào trong hệ thống.

+ Người dùng chọn nút “Lưu”. (Luồng A2).

+ Hệ thống kiểm tra dữ liệu nhập vào. (Luồng A3).

+ Sửa thành công, kết thúc use case.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1:

+ Người dùng không muốn thêm học sinh mới, ấn nút “Không” trong thông báo xác nhận thông tin thêm học sinh.

+ Thêm không thành công, kết thúc use case.

- Luồng A2:

+ Người dùng không muốn thay đổi thông tin học sinh, ấn nút “Không” trong thông báo xác nhận thông tin sửa đổi học sinh.

+ Thay đổi không thành công, kết thúc use case.

- Luồng A3:

+ Hệ thống kiểm tra dữ liệu nhập vào có sai sót, đưa ra thông báo “Lưu không thành công”. Yêu cầu nhập lại.

c) Tiền điều kiện: Tác nhân phải đăng nhập vào hệ thống.

d) Hậu điều kiện: Nếu ca sử dụng được thực hiện thành công thì thông tin học sinh sẽ được thêm, sửa, còn lại các trường hợp khác thông tin vẫn được giữ nguyên.

### 3.2.3. Đặc tả Usecase quản lý môn học

a) Mô tả tóm tắt

- Tên UC: Quản lý thông tin môn học

- Mục đích: Mô tả cách người quản lý thêm, sửa, xóa thông tin môn học

- Tác nhân: Giáo viên.

b) Các Luồng sự kiện

**Luồng sự kiện chính:**

- Từ màn hình làm việc chính chọn chức năng Quản lý thông tin môn học

- Hệ thống hiện thị các lựa chọn

+ Thêm môn học

+ Sửa môn học

+ Xóa môn học

- Thêm môn học:

+ Người quản lý muốn thêm môn học mới vào trong hệ thống.

+ Người quản lý chọn nút “Thêm mới”

+ Hệ thống yêu cầu nhập thông tin cần thiết.

+ Người quản lý nhập thông tin cần thiết vào trong hệ thống.

+ Người quản lý chọn nút “Lưu” (Luồng A1).

+ Hệ thống kiểm tra dữ liệu nhập vào (Luồng A3).

+ Thêm thành công, kết thúc use case.

+ Kết quả: Một môn học mới được thêm vào.

- Sửa thông tin môn học.

+ Người quản lý chọn môn học cần sửa thông tin.

+ Người quản lý nhập thông tin cần thay đổi vào trong hệ thống.

+ Người quản lý chọn nút “Lưu” (Luồng A2).

+ Hệ thống kiểm tra dữ liệu nhập vào (Luồng A3).

+ Sửa thành công, kết thúc use case.

- Xóa môn học

+ Người quản lý chọn môn học cần xóa.

+ Người quản lý chọn nút “Xóa” (Luồng A4).

+ Hệ thống kiểm tra dữ ràng buộc dữ liệu (Luồng A5).

+ Xóa thành công, kết thúc use case.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1:

+ Người quản lý không muốn thêm môn học mới, ấn nút “Không” trong thông báo xác nhận thông tin thêm môn học.

+ Thêm không thành công, kết thúc use case.

- Luồng A2:

+ Người quản lý không muốn thay đổi thông tin môn học, ấn nút “Không” trong thông báo xác nhận thông tin sửa đổi môn học.

+ Thay đổi không thành công, kết thúc use case.

- Luồng A3:

+ Hệ thống kiểm tra dữ liệu nhập vào có sai sót, đưa ra thông báo “Lưu không thành công”. Yêu cầu nhập lại.

- Luồng A4:

+ Người quản lý không muốn xóa môn học, ấn nút “Không” trong thông báo xác nhận xóa môn học.

+ Xóa không thành công, kết thúc use case.

- Luồng A5:

+ Hệ thống kiểm tra ràng buộc dữ liệu môn học không xóa được, đưa ra thông báo “không thể xóa”. Use case xóa kết thúc

c) Tiền điều kiện: Tác nhân phải đăng nhập vào hệ thống.

d) Hậu điều kiện: Nếu ca sử dụng được thực hiện thành công thì thông tin môn học sẽ được thêm, sửa hoặc xóa, còn lại các trường hợp khác thông tin vẫn được giữ nguyên.

### 3.2.4. Đặc tả Usecase quản lý Điểm

a) Mô tả tóm tắt

- Tên UC: Quản lý điểm

- Mục đích:

+ Mô tả việc tác nhân cập nhật điểm của học sinh sau mỗi học kỳ, mỗi năm học.

+ Mô tả tác nhân đã chọn một trong các chức năng nhập điểm, sửa điểm hoặc xóa điểm sau đó giáo viên có thể thêm, sửa, xóa điểm của học sinh và kết thúc.

- Tác nhân: Giáo viên

b) Các luồng sự kiện

**Luồng sự kiện chính:**

- Ca sử dụng bắt đầu khi giáo viên đăng nhập hệ thống thành công.

- Người quản lý mở chức năng nhập điểm.

- Hệ thống nhắc giáo viên chọn môn học, giáo viên chọn môn học, hệ thống nhắc giáo viên chọn việc:

+ Nhập điểm

+ Sửa điểm

+ Tính điểm trung bình

- Nhập điểm:

+ Hệ thống hiển thị màn hình danh sách các môn học mà giáo viên giảng dạy trong học kỳ gồm tên môn học, mã số môn học.

+ Giáo viên chọn một môn học, hệ thống hiển thị danh sách các lớp học mà giáo viên đó dạy.

+ Giáo viên chọn một lớp học để nhập điểm và tiến hành nhập điểm.

+ Khi nhập điểm xong giáo viên nhấn vào nút “Lưu”

+ Hệ thống tiến hành kiểm tra điểm vừa nhập. Nếu nhập sai thực hiện luồng A1

+ Hệ thống tiến hành lưu điểm giáo viên vừa nhập.

+ Kết thúc, hệ thống hiển thị thông báo nhập điểm thành công.

- Sửa điểm:

+ Hệ thống hiển thị màn hình danh sách các môn học mà giáo viên giảng dạy trong học kỳ gồm tên môn học, mã số môn học.

+ Giáo viên chọn một môn học, hệ thống hiển thị danh sách các lớp học mà giáo viên đó dạy.

+ Giáo viên chọn một lớp học cần sửa điểm và tiến hành sửa điểm.

+ Khi sửa điểm xong giáo viên click và nút lưu, hệ thống tiến hành lưu điểm giáo viên vừa sửa.

+ Hệ thống tiến hành kiểm tra điểm vừa nhập. Nếu sai thực hiện luồng A2

+ Hệ thống tiến hành lưu điểm giáo viên vừa sửa.

+ Kết thúc, hệ thống hiển thị thông báo sửa điểm thành công.

- Tính điểm trung bình:

+ Hệ thống hiển thị màn hình danh sách các môn học mà giáo viên giảng dạy trong học kỳ gồm tên môn học, mã số môn học.

+ Giáo viên chọn một môn học, hệ thống hiển thị danh sách các lớp học mà giáo viên đó dạy.

+ Giáo viên chọn một lớp học cần tính điểm trung bình và tiến hành tính điểm.

+ Khi tính điểm xong giáo viên click và nút lưu, hệ thống tiến hành lưu lại bảng điểm mới.

+ Kết thúc, hệ thống hiển thị thông báo tính điểm thành công.

+ Sau khi Giáo viên thêm, sửa hoặc tính điểm xong, Người quản lý khóa chức năng nhập điểm.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1: Nhập điểm sai định dạng

+ Hệ thống thông báo: Điểm phải nhập từ 1-> 10, làm tròn đến chữ số thập phân thứ 2 và không được để trống.

+ Giáo viên tiến hành nhập lại điểm.

- Luồng A2: Giáo viên sửa điểm sai định dạng hoặc để điểm trống.

+ Hệ thống thông báo: Điểm phải nhập từ 1-> 10, làm tròn đến chữ số thập phân thứ 2 và không được để trống.

+ Giáo viên tiến hành sửa lại điểm.

c) Tiền điều kiện: Giáo viên cần đăng nhập hệ thống

d) Hậu điều kiện: Nếu ca sử dụng được thực hiện thành công thì thông tin điểm của học sinh sẽ được nhập, sửa hoặc tính điểm trung bình, còn lại các trường hợp khác thông tin vẫn được giữ nguyên.

### 3.2.5. Đặc tả Usecase thống kê

a) Mô tả tóm tắt

- Tên UC: Thống kê

- Mục đích: Thống kê một danh sách theo mục đích của người dùng.

- Tác nhân: Giáo viên, Người quản lý

b) Các luồng sự kiện

- Ca sử dụng bắt đầu khi người dùng đăng nhập hệ thống thành công

- Từ giao diện chính chọn chức năng thống kê.

- Hệ thống hiển thị biểu mẫu thống kê

- Chọn mục đích thống kê theo yêu cầu người dùng. Nhấn nút “Xem”.

- Hiển thị danh sách cần thống kê. Kết thúc use case thống kê.

c) Tiền điều kiện: Người dùng cần đăng nhập hệ thống hợp lệ

d) Hậu điều kiện: Ca sử dụng được thực hiện thành công người dùng có thể xem được danh sách thống kê theo yêu cầu.

### 3.2.6. Đặc tả Usecase tra cứu thông tin

a) Mô tả tóm tắt

- Tên UC: Tra cứu thông tin

- Mục đích: Tra cứu thông tin theo mục đích của người dùng..

- Tác nhân: Giáo viên, học sinh

b) Các luồng sự kiện

**Luồng sự kiện chính:**

-Từ giao diện chính chọn chức năng “Tra cứu”.

- Hệ thống hiển thị giao diện tra cứu.

- Người dùng nhập thông tin cần tra cứu và chọn nút “Tìm”. Nếu không muốn tìm kiếm nữa thì thực hiện luồng A1.

- Hệ thống hiển thị kết quả tìm kiếm.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1: Muốn thoát

+ Người dùng bấm nút Hủy để kết thúc tìm kiếm thông tin.

+ Hệ thống sẽ kết thúc việc tìm kiếm thông tin

c) Tiền điều kiện: Người dùng cần đăng nhập hệ thống hợp lệ

d) Hậu điều kiện: Ca sử dụng được thực hiện thành công người dùng có thể xem được kết quả tìm kiếm theo yêu cầu.

### 3.2.7. Đặc tả Usecase quản lí giáo viên:

a) Mô tả tóm tắt

- Tên UC: Quản lý giáo viên

- Mục đích: Mô tả cách người quản lý thêm, sửa, xóa giáo viên

- Tác nhân: Người quản lí (Admin)

b) Các Luồng sự kiện

**Luồng sự kiện chính:**

- Từ màn hình làm việc chính của hệ thống cho người quản lí chọn chức năng quản lí giáo viên

- Hệ thống hiện thị các lựa chọn

+ Thêm giáo viên

+ Sửa giáo viên

+ Xóa giáo viên

+ Tìm kiếm giáo viên

- Thêm giáo viên:

+ Người quản lý muốn thêm giáo viên mới vào trong hệ thống.

+ Người quản lý chọn nút “Thêm mới”

+ Hệ thống yêu cầu nhập thông tin cần thiết.

+ Người quản lý nhập thông tin cần thiết vào trong hệ thống.

+ Người quản lý chọn nút “Lưu” (Luồng A1).

+ Hệ thống kiểm tra dữ liệu nhập vào (Luồng A3).

+ Thêm thành công, kết thúc use case.

+ Kết quả: Một giáo viên mới được thêm vào.

- Sửa thông tin giáo viên.

+ Người quản lý chọn giáo viên cần sửa thông tin.

+ Người quản lý nhập thông tin cần thay đổi vào trong hệ thống.

+ Người quản lý chọn nút “Lưu” (Luồng A2).

+ Hệ thống kiểm tra dữ liệu nhập vào (Luồng A3).

+ Sửa thành công, kết thúc use case.

- Xóa giáo viên

+ Người quản lý chọn giáo viên cần xóa khỏi hệ thống.

+ Người quản lý chọn nút “Xóa” (Luồng A4).

+ Hệ thống kiểm tra dữ ràng buộc dữ liệu (Luồng A5).

+ Xóa thành công , kết thúc use case.

- Tìm kiếm giáo viên

+ Người quản lý nhấn vào thanh tìm kiếm trong mục giáo viên.

+ Người quản lý gõ kết quả và nhấn vào biểu tượng tìm kiêmd

+ Hệ thống kiểm tra dữ liệu ràng buộc và truy xuất kết quả tìm kiếm.

+ Tìm kiếm thành công, kết thúc usecse.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1:

+ Người quản lý không muốn thêm giáo viên mới, ấn nút “Không” trong thông báo xác nhận thông tin thêm môn học.

+ Thêm không thành công, kết thúc use case.

- Luồng A2:

+ Người quản lý không muốn thay đổi thông tin giáo viên, ấn nút “Không” trong thông báo xác nhận thông tin sửa đổi môn học.

+ Thay đổi không thành công, kết thúc use case.

- Luồng A3:

+ Hệ thống kiểm tra dữ liệu nhập vào có sai sót, đưa ra thông báo “Lưu không thành công”. Yêu cầu nhập lại.

- Luồng A4:

+ Người quản lý không muốn xóa giáo viên, ấn nút “Không” trong thông báo xác nhận xóa môn học.

+ Xóa không thành công, kết thúc use case.

- Luồng A5:

+ Hệ thống kiểm tra ràng buộc dữ liệu giáo viên không xóa được, đưa ra thông báo “không thể xóa”. Use case xóa kết thúc

c) Tiền điều kiện: Tác nhân phải đăng nhập vào hệ thống.

d) Hậu điều kiện: Nếu ca sử dụng được thực hiện thành công thì thông tin môn học sẽ được thêm, sửa hoặc xóa, còn lại các trường hợp khác thông tin vẫn được giữ nguyên.

### 3.2.8. Đặc tả Usecase quản lý và hiệu chỉnh tài khoản.

a) Mô tả tóm tắt

- Tên UC: Quản lý và hiệu chỉnh tài khoản

- Mục đích: Mô tả người quản lý thêm, sửa, xóa thông tin học sinh

- Tác nhân: Giáo viên.

b) Các luồng sự kiện

**Luồng sự kiện chính:**

- Từ màn hình làm việc chính của hệ thống người quản lí chọn chức năng Quản lý và hiệu chỉnh tài khoản

- Hệ thống hiện thị các lựa chọn

+ Thêm tài khoản

+ Sửa tài khoản

+ Xóa tài khoản

- Thêm tài khoản:

+ Người quản lí muốn thêm tài khoản mới vào trong hệ thống để đăng nhập.

+ Người quản lí chọn nút “Thêm mới”.

+ Hệ thống yêu cầu nhập thông tin cần thiết.

+ Người dùng nhập thông tin cần thiết vào trong hệ thống.

+ Người dùng chọn nút “Lưu”(Luồng A1).

+ Hệ thống kiểm tra dữ liệu nhập vào(Luồng A3).

+ Thêm thành công, kết thúc use case.

+ Kết quả: Một tài khoản mới được thêm vào.

- Sửa thông tin tài khoản.

+ Người quản lí chọn tài khoản cần sửa thông tin.

+ Người quản lí nhập thông tin cần thay đổi vào trong hệ thống.

+ Người quản lí chọn nút “Lưu”. (Luồng A2).

+ Hệ thống kiểm tra dữ liệu nhập vào. (Luồng A3).

+ Sửa thành công, kết thúc use case.

**Luồng sự kiện rẽ nhánh:**

- Luồng A1:

+ Người quản lí không muốn thêm tài khoản mới, ấn nút “Không” trong thông báo xác nhận thông tin thêm tài khoản.

+ Thêm không thành công, kết thúc use case.

- Luồng A2:

+ Người quản lí không muốn thay đổi thông tin tài khoản, ấn nút “Không” trong thông báo xác nhận thông tin sửa đổi tài khoản.

+ Thay đổi không thành công, kết thúc use case.

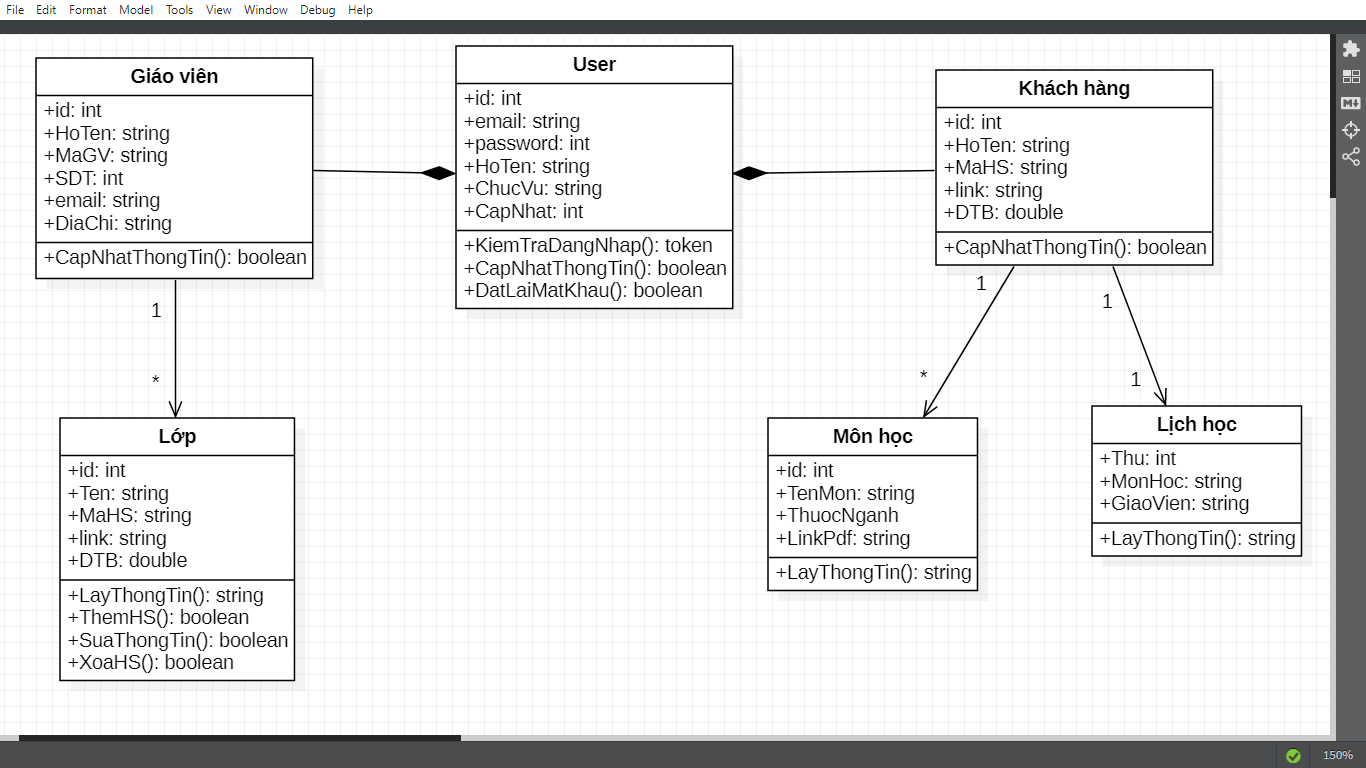
- Luồng A3:

+ Hệ thống kiểm tra dữ liệu nhập vào có sai sót, đưa ra thông báo “Lưu không thành công”. Yêu cầu nhập lại.

c) Tiền điều kiện: Tác nhân phải đăng nhập vào hệ thống.

d) Hậu điều kiện: Nếu ca sử dụng được thực hiện thành công thì thông tin tài khoản sẽ được thêm, sửa, xóa còn lại các trường hợp khác thông tin vẫn được giữ nguyên.

## 3.3. Biểu đồ lớp



***Hình 3 - 11: Biểu đồ lớp***

# CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG

## 4.1. Hệ thống CSDL

Gồm 6 bảng chính là những bảng sau: user, giáo viên, học sinh, lớp, lịch học, môn học

### 4.1.1. Bảng user

Truy vấn: INSERT INTO `user` (`id`, `email`, `password`, `name`) VALUES ('1', 'oanh@gmail.com', '12345678', '12345678');

| Cột | Kiểu | Null |
| --- | --- | --- |
| id *(Chính)* | int(11) | Không |
| email | varchar(100) | Không |
| password | int(11) | Không |
| name | int(11) | Không |

***Bảng 4 - 1: Bảng user***

### 4.1.2. Bảng giáo viên

Truy vấn : INSERT INTO `giaovien` (`id`, `name`, `magv`, `sdt`, `email`, `diachi`) VALUES ('1', 'Nguyễn Mai Phương Anh', '103AB', '0300000002', 'phuonganh@gmail.com', '123 Lý Thường Kiệt, Bắc Lý, Đồng Hới');

| Cột | Kiểu | Null |
| --- | --- | --- |
| id *(Chính)* | int(11) | Không |
| name | varchar(50) | Không |
| magv | varchar(50) | Không |
| sdt | int(11) | Không |
| email | varchar(100) | Không |
| diachi | varchar(255) | Không |

***Bảng 4 - 2: Bảng giáo viên***

### 4.1.3. Bảng học sinh

Truy vấn: INSERT INTO `hocsinh` (`id`, `name`, `mahs`, `link`, `DTB`) VALUES (7, 'Trần Thị Kim Oanh', '19it184', 'http://192.168.1.105/Web/Anhthe/anhthe4.jpg', '8');

| Cột | Kiểu | Null |
| --- | --- | --- |
| id *(Chính)* | int(50) | Không |
| name | varchar(50) | Không |
| mahs | varchar(50) | Không |
| link | varchar(100) | Không |
| DTB | double | Không |

***Bảng 4 - 3: Bảng học sinh***

### 4.1.4. Bảng lớp

Truy vấn: INSERT INTO `lop` (`id`, `name`, `mahs`, ` link `, ` DTB `) VALUES ('1', 'Lop1A', '19it184', ''http://192.168.1.105/Web/Anhthe/anhthe4.jpg ', '8',);

***Bảng 4 - 4: Bảng lớp***

| Cột | Kiểu | Null |
| --- | --- | --- |
| id *(Chính)* | int(11) | Không |
| name | varchar(50) | Không |
| mahs | varchar(50) | Không |
| link | varchar(100) | Không |
| DTB | double | Không |

### 4.1.5. Bảng lịch học

Truy vấn: INSERT INTO `lichhoc` (`thu`, `monhoc`, `giaovien`) VALUES ('2', 'Toán', 'Nguyễn Mai Phương Anh');

***Bảng 4 - 5: Bảng lịch học***

| Cột | Kiểu | Null |
| --- | --- | --- |
| thu *(Chính)* | int(11) | Không |
| monhoc | varchar(100) | Không |
| giaovien | varchar(100) | Không |

### 4.1.6. Bảng môn học

Truy vấn:INSERT INTO `monhoc` (`id`, `tenmon`, `thuocnganh`, `linkpdf`) VALUES ('4', 'toan', 'toan hoc', 'https://docs.google.com/document/d/1\_H04lyR0aiGzQTpzi3Ja2corSiqI-ByY4ezlSEHUKHU/edit');

***Bảng 4 - 6: Bảng môn học***

| Cột | Kiểu | Null |
| --- | --- | --- |
| id *(Chính)* | int(50) | Không |
| tenmon | varchar(50) | Không |
| thuocnganh | varchar(50) | Không |
| linkpdf | varchar(100) | Không |

## 4.2. Thiết kế ứng dụng

### 4.2.1. Phần web API

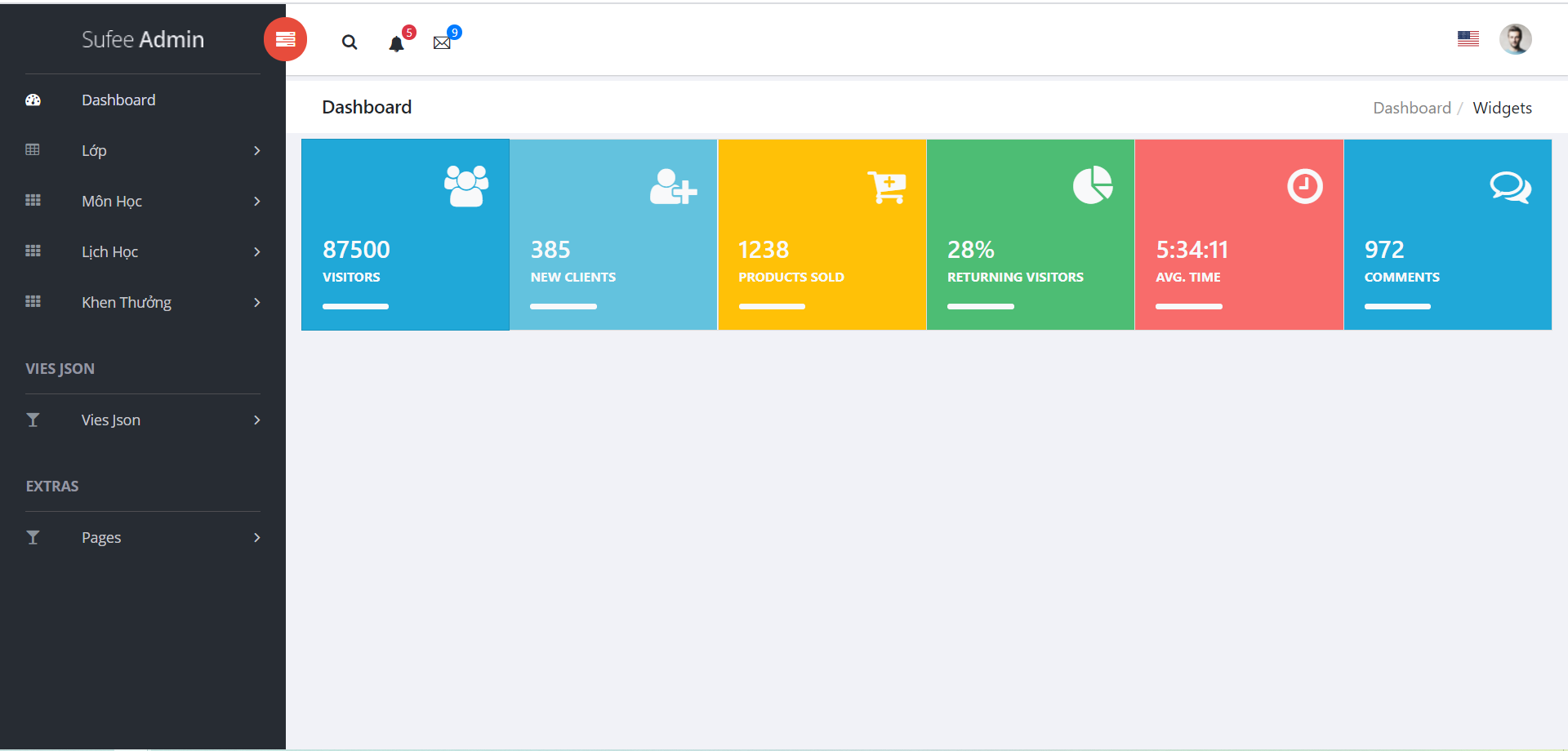
Trang chủ giao diện quản lý học sinh những phần chính như lớp, môn học, lịch học và khen thưởng.

- Lớp: Hiện thị danh sách học sinh theo từng lớp. Tại phần này người quản lý có thể thêm học sinh vào danh sách, chỉnh sửa thông tin của học sinh đã có trong danh sách và xóa học sinh khỏi danh sách.

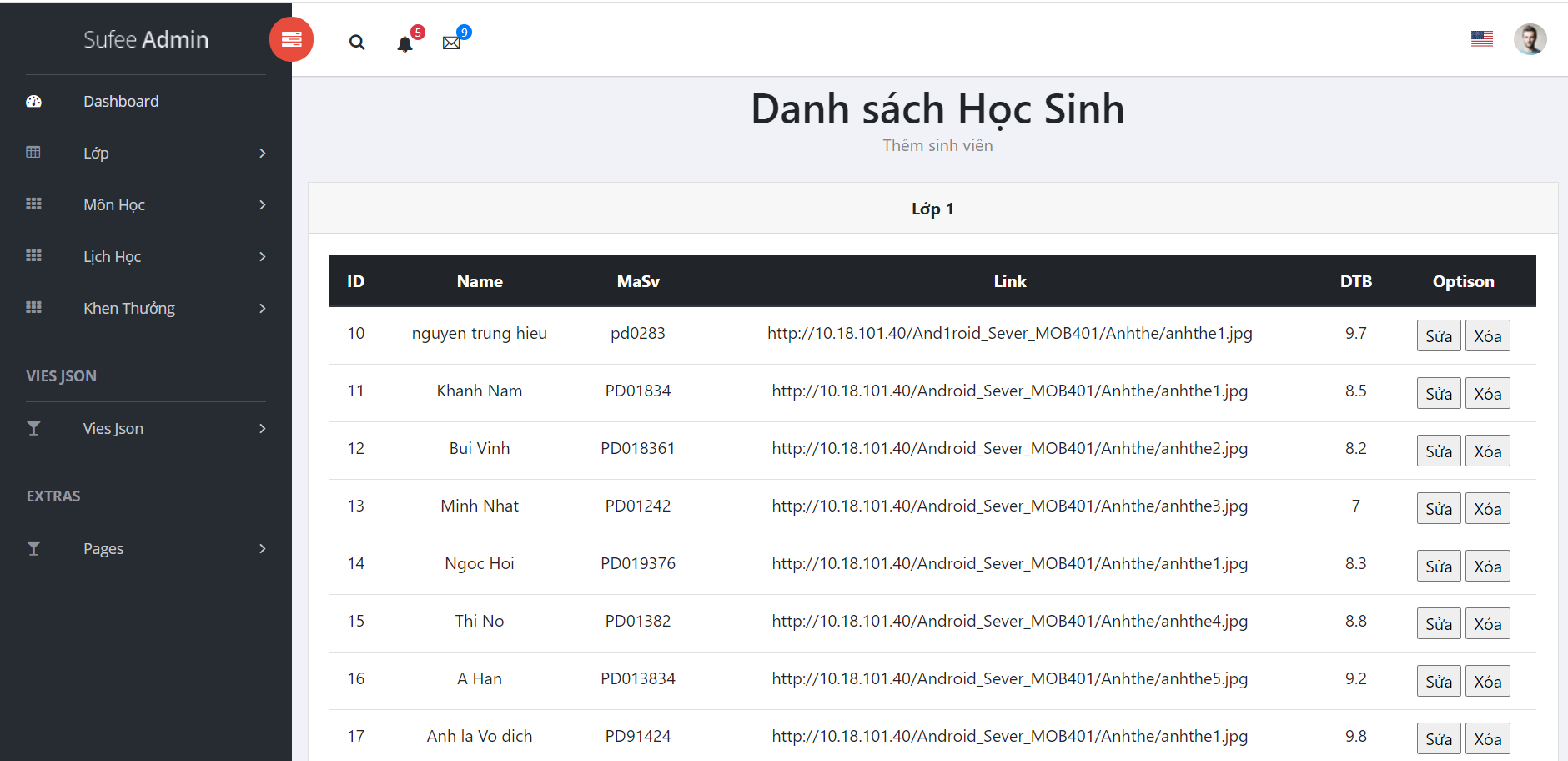
- Môn học: Hiện thị danh sách môn học. Danh sách các môn học có thể thay đổi, thêm một môn học mới vào danh sách, chỉnh sửa môn học đã có và xóa môn học,

- Lịch học: Hiện thị lịch học theo ngày học. Tại đây người quản lý có thể chỉnh sửa, cập nhật và tạo mới lịch học.

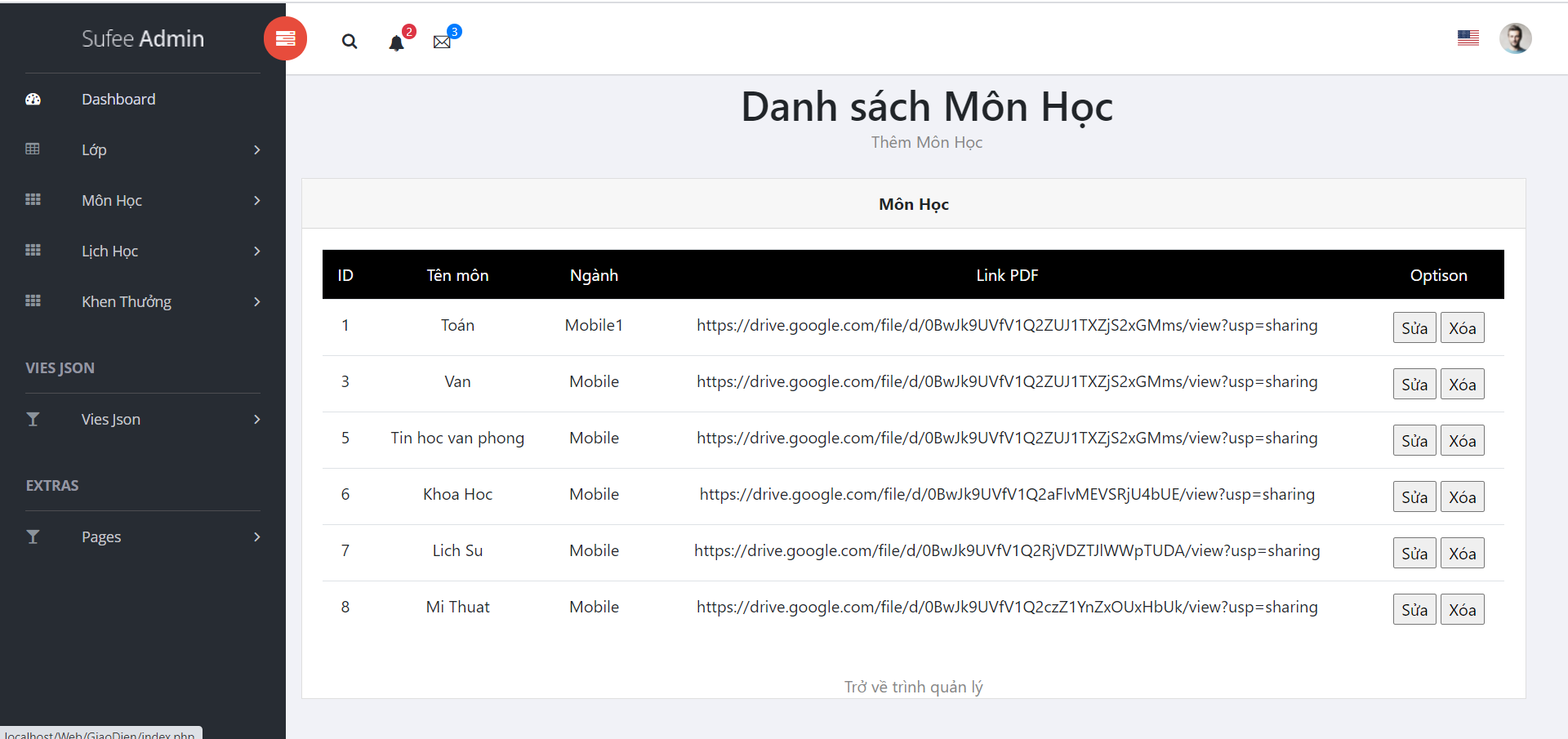
- Khen thưởng: Hiển thị danh sách khen thưởng. Danh sách khen thưởng được chọn và đánh giá dựa trên điểm trung bình được cập nhật ở phần danh sách học sinh theo lớp. Những học sinh có điểm trung bình trên 8 sẽ được xét duyệt học bổng.



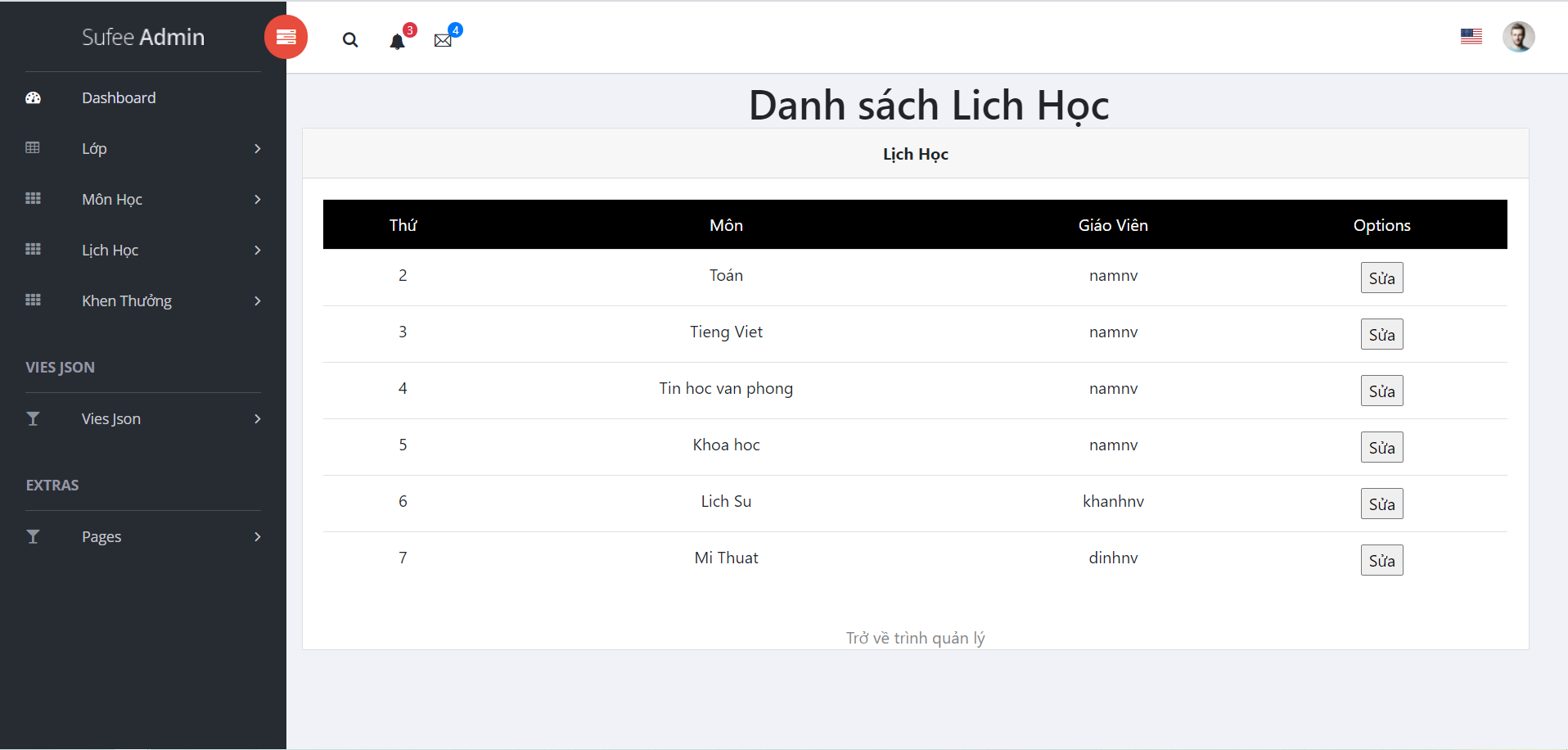
***Hình 4 - 1: Trang chủ website***



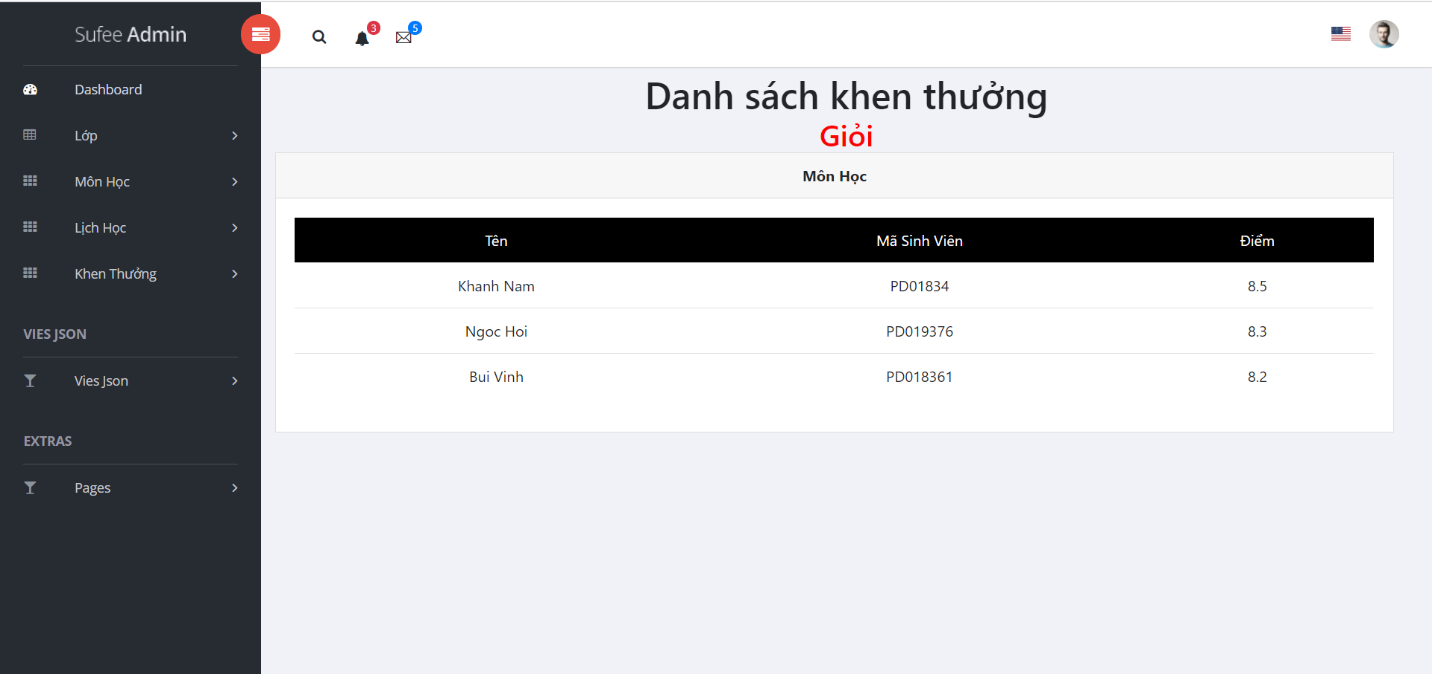
***Hình 4 - 2: Danh sách học sinh xếp theo lớp***



***Hình 4 - 3: Danh sách môn học***



***Hình 4 - 4: Danh sách lịch học***



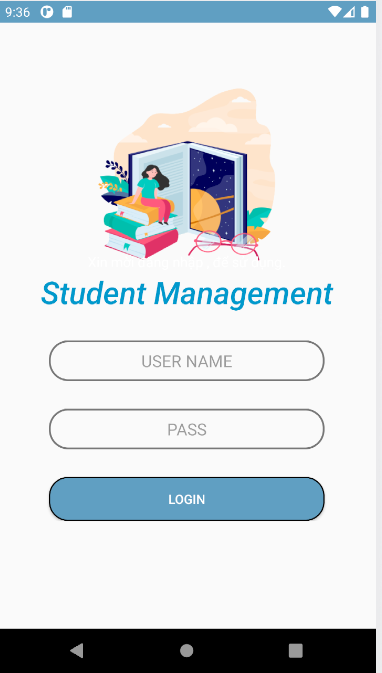
***Hình 4 - 5: Danh sách khen thưởng***

### 4.2.2. Phần ứng dụng Android

Ứng dụng dụng được xây dựng dựa trên nền tảng web API. Toàn bộ dữ liệu của ứng dụng đều được Json trả về .

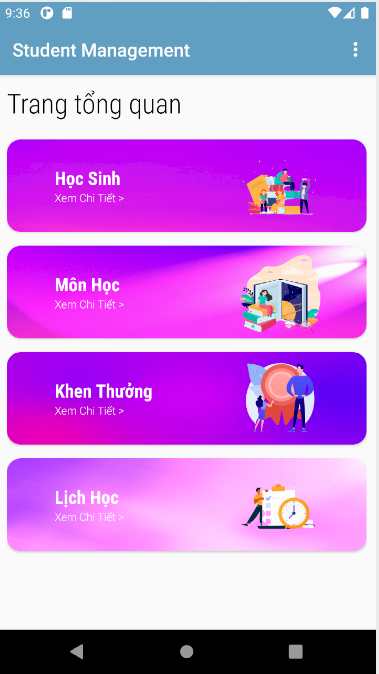
Một số giao diện và chức năng của ứng dụng:

- Phần đăng nhập: Người dùng chỉ đăng nhập khi được cấp tài khoản, với những trường hợp không có tài khoản và không đăng nhập được thì hệ thống sẽ thông báo đăng nhập không hợp lệ. Mọi dữ liệu đăng nhập đều được lưu trữ và truy xuất trong hệ quản trị cơ sở dữ liệu.



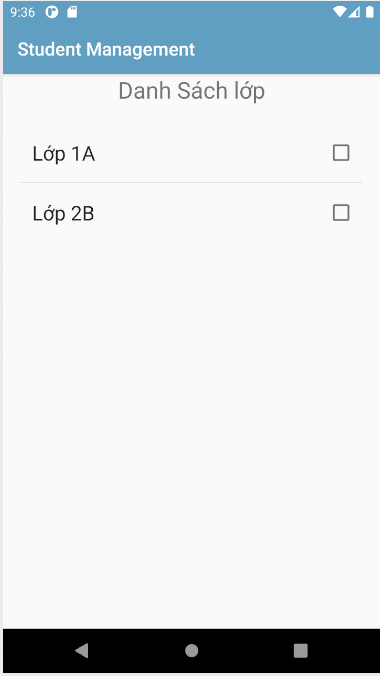
***Hình 4 - 6: Giao diện đăng nhập ứng dụng***

- Phần trang chủ: Trang chủ ứng dụng là nơi hiển thị đầy đủ các phân mục của ứng dụng. Với hiện thi trực quan dễ sử dụng.

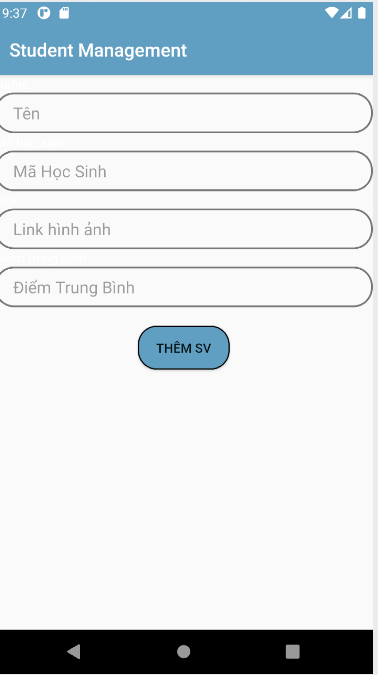


***Hình 4 - 7: Giao diện trang chủ ứng dụng***

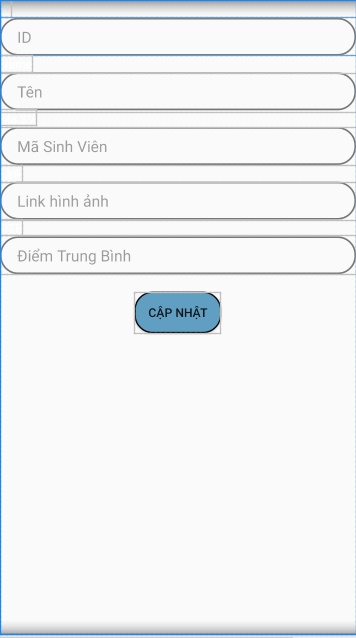
-Phần Học Sinh: Hiển thị danh sách học sinh theo lớp. Thêm một học sinh mới vào danh sách, chỉnh sửa cập nhật lại thông tin của học sinh, xóa học sinh ra khỏi danh sách

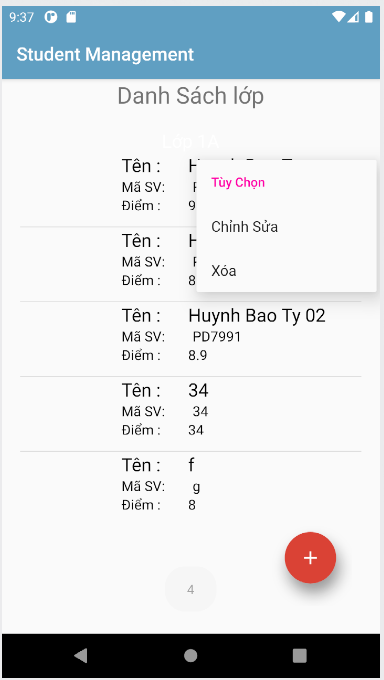
***Hình 4 - 8: Giao diện danh sách học sinh phân theo lớp***



***Hình 4 - 9: Thêm học sinh***

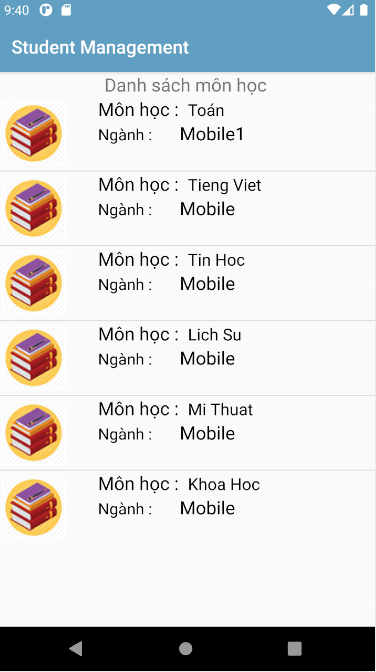


***Hình 4 - 10: Chỉnh sửa và cập nhật thông tin học sinh***



***Hình 4 - 11: Xóa học sinh ra khỏi danh sách***

- Phần Môn Học: Hiện thị danh sách môn học theo ngành học.



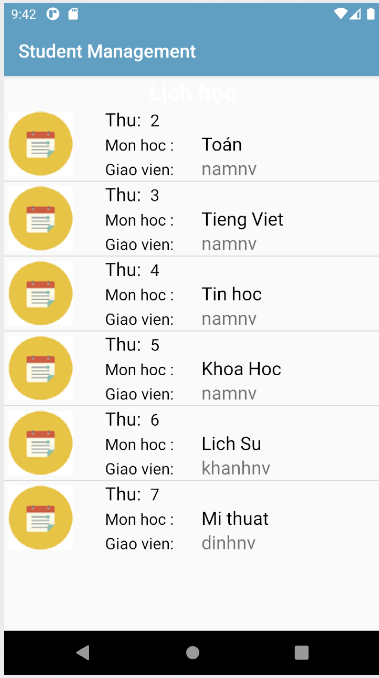
***Hình 4 - 12: Danh sách môn học***

- Phần khen thưởng: Hiện thị thông tin của những học sinh đạt thành tích cao. Hệ thống này được xác lập dựa trên điểm trung bình ở danh sách học sinh,



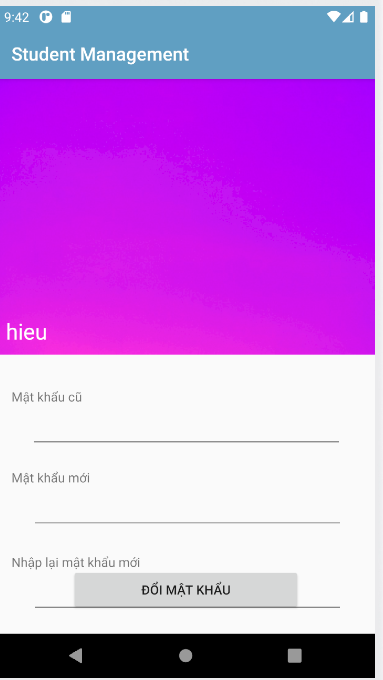
***Hình 4 - 13: Danh sách khen thưởng***

- Phần lịch học:



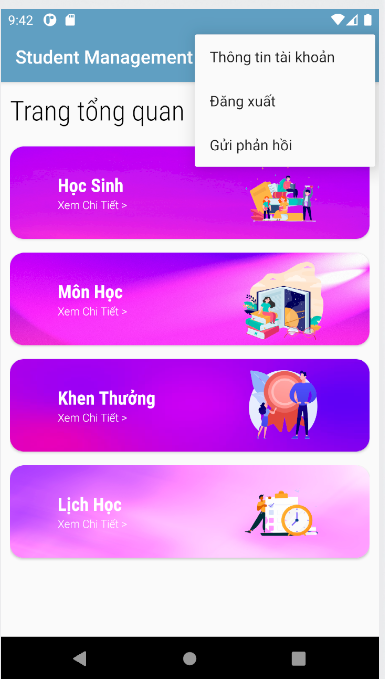
***Hình 4 - 14: Giao diện lịch học***

- Phần thay đổi mật khẩu người dùng:



***Hình 4 - 15: Giao diện thay đổi mật khẩu người dùng***

- Phần thông tin tài khoản và đăng xuất:



***Hình 4 - 16: Giao diện thông tin tài khoản, đăng xuất***

# KẾT LUẬN

## Kết quả đạt được

***Về kiến thức học tập***

Để xây dựng một hệ thống quản lý nói chung và quản lý học sinh nói riêng là một việc không hề đơn giản, nó đòi hỏi chúng ta phải xây dựng theo các giai đoạn: khảo sát, phân tích và thiết kế hệ thống, nghiên cứu tổng quan các công cụ lập trình, thiết kế phần mềm và kiểm thử. Ngày nay chúng ta đang đứng trước nhiều lựa chọn về công nghệ mới vậy nên một ứng dụng giúp quản lý học sinh một cách đơn giản, tối ưu hơn là điều không thể thiếu. Thông qua nghiên cứu và xây dựng đề tài này, chúng em đã củng cố, học hỏi thêm được nhiều thứ như:

- Củng cố kiến thức về lập trình web bằng ngôn ngữ php, laravel framework. Cũngbiết thêm được nhiều kiến thức về quản lý cũng như phân tích hệ thống của một ứng dụng quản lý.

- Trau dồi kiến thức và kĩ năng lập trình di động, Từ đó có thể phát triển và xây dựngnên một số hệ thống quản lí tương tự.

- Cũng cố kiến thức về phân tích thiết kế hệ thống UML, vẽ và thao tác thành thạo các loại biểu đồ cơ bản như: biểu đồ use-case, biểu đồ lớp, biểu đồ tuần tự… Đặc tả được use-case.

- Biết cách viết và sử dụng API để lấy dữ liệu từ web qua ứng dụng đây là một công nghệ rất hay nó giúp chúng ta tiết kiệm được nhiều thời gian hơn, tối ưu về mặt quản trị hệ CSDL.

***Về phần mềm***

Đứng trước xu hướng phát triển của công nghệ thông tin như vũ bão, việc tin học hoá việc quản lý học sinh là vô cùng quan trọng và bức thiết. Nó sẽ giúp cho các nhà trường cũng như các giáo viên quản lý và quan sát một cách tối ưu về thành tích học tập cũng như thông tin học sinh. Qua dự án trên nhóm chúng em đạt được một số kết quả như sau:

- Xây dựng được hệ thống web API và ứng dụng android với giao diện phù hợp, chưa được tối ưu hoàn toàn nhưng đủ bắt mắt với người dùng.

- Ứng dụng có một số chức năng căn bản của một ứng dụng quản lý cần có như quản lý danh sách học sinh, quản lý danh sách khen thưởng, điểm học sinh, quản lý môn học, lịch học…

- Ứng dụng giúp người dùng tối ưu hóa dữ liệu người dùng, cơ sở dữ liệu rõ ràng liên kết hoàn toàn trên web API.

**Hạn chế**

Dự án quản lý học sinh tiểu học tuy chưa được hoàn thiện hoàn toàn nhưng nó cũng đã có những chức năng căn bản của một ứng dụng. Do thời gian hoàn thiện dự án còn hạn chế nên một số chức năng khác nhóm chúng em chưa kịp hoàn thiện trong thời gian trước khi bảo vệ, nên dự án vẫn có một số chức năng chưa kịp hoàn thiện mong thầy cô giáo châm chước nhiều. Song nếu có thời gian nhiều hơn nhóm chúng em sẽ hoàn thiện và bảo trì dự án tốt hơn từ đó có thể đưa ứng dụng vào thực tiễn.

## Hướng phát triển

Để dự án có thể hoạt động hiệu quả hơn cần dữ liệu phải đượ clưu trữ bằng hệ quản trị cơ sở dữ liệu có tính bảo mật hơn. Cần có sự kết hợp song song giữa nhà trường và giáo viên một cách chặt chẽ hơn để việc theo dõi và quản lý học sinh được tốt hơn. Nếu có thể đưa dự án quản lý học sinh này vào thực tiễn thì cần sự phối hợp chặt chẽ giữa nhà trường, giáo viên và phụ huynh học sinh. Dự án sẽ được hoàn thiện trong thời gian tới và sẽ được ứng dụng vào thực tế nếu nó mang tính thuyết phục cao. Vì vậy trong khuôn khổ đồ án này chúng em chỉ có thể đưa ra những chức năng cơ bản, mô phỏng một số yêu cầu trong thực tế. Đồ án của chúng em còn rất nhiều thiếu sót, rất mong được sự đóng góp ý kiến từ quý thầy cô để chúng em hoàn thiện dự án này hơn.

# TÀI LIỆU THAM KHẢO

[1] Tìm hiểu kiến thức lập trình ứng dụng:

[Online] developer.android.com

[2] Tìm hiểu kiến thức xây dựng website và ứng dụng:

[Online] w3schools.com

[3] Tìm hiểu về ngôn ngữ thiết kế hệ thống UML:

[Online] [staruml.io/](https://staruml.io/)

[4] Vai trò của công cụ lập trình Sublime Text

[Online] [lamvt.vn/sublime-text-la-gi-vi-sao-no-quan-trong-voi-cac-lap-trinh-vien/](https://lamvt.vn/sublime-text-la-gi-vi-sao-no-quan-trong-voi-cac-lap-trinh-vien/)

[5] Tìm hiểu cách phân tích và vẽ biểu đồ UML

[Online] [viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-2- 0bDM6wpAG2X4](https://viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-2-0bDM6wpAG2X4)

[6] Tìm hiểu các thông tin cần thiết cho đồ án:

[Online] [www.google.com.vn](http://www.google.com.vn)

[7] Cách thức hoạt động API

[Online]blogs.mulesoft.com/learn-apis/api-led-connectivity/what-are-apis-how-do-apis-work

[Online][careers.upwork.com/blogs/startup-resources-upwork-blog/what-is-an-api-and-how-does-it-work](https://careers.upwork.com/blogs/startup-resources-upwork-blog/what-is-an-api-and-how-does-it-work)